



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,
Comércio e Serviços

Instituto Nacional da Propriedade Industrial



(11) BR 112017001975-2 B1

(22) Data do Depósito: 29/07/2015

(45) Data de Concessão: 28/02/2023

(54) Título: IMAGEAMENTO COMPUTACIONAL DE BAIXA POTÊNCIA

(51) Int.Cl.: G06F 1/32; G06T 1/20.

(30) Prioridade Unionista: 12/08/2014 US 14/458,052; 12/08/2014 US 14/458,014; 30/07/2014 US 62/030,913.

(73) Titular(es): MOVIDIUS LIMITED.

(72) Inventor(es): BRENDAN BARRY; DAVID MOLONEY; FERGAL CONNOR; RICHARD RICHMOND'.

(86) Pedido PCT: PCT IB2015001845 de 29/07/2015

(87) Publicação PCT: WO 2016/016730 de 04/02/2016

(85) Data do Início da Fase Nacional: 30/01/2017

(57) Resumo: ?IMAGEAMENTO COMPUTACIONAL DE BAIXA POTÊNCIA? O presente pedido de patente revela um dispositivo de computação (100) que pode fornecer uma plataforma de computação de baixa potência, altamente capaz para imageamento computacional. O dispositivo de computação pode incluir uma ou mais unidades de processamento, por exemplo um ou mais processadores vetoriais (102) e um ou mais aceleradores de hardware (104), um tecido de memória inteligente (106), um dispositivo periférico (108) e um módulo de gerenciamento de energia (110). O dispositivo de computação pode se comunicar com dispositivos externos, tais como um ou mais sensores de imagem, um acelerômetro, um giroscópio ou qualquer outro dispositivo de sensor apropriado.

"IMAGEAMENTO COMPUTACIONAL DE BAIXA POTÊNCIA"**REFERÊNCIA REMISSIVA A PEDIDOS RELACIONADOS**

[0001] Este pedido reivindica o benefício da prioridade de data anterior do Pedido de Patente U.S. No. 14/458.014, intitulado "LOW POWER COMPUTATIONAL IMAGING" depositado em 12 de Agosto de 2014, o qual reivindica a prioridade do Pedido de Patente U.S. Provisório No. 62/030.913, intitulado "LOW POWER COMPUTATIONAL IMAGING COMPUTING DEVICE" depositado em 30 de Julho de 2014 e do Pedido de Patente U.S. No. 14/082.396, intitulado "APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING COMPUTATIONAL IMAGING PIPELINE" depositado em 18 de Novembro de 2013, que reivindica a prioridade do Pedido de Patente Romeno OSIM Registratura A/00812, intitulado "APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE" depositado em 6 de Novembro de 2013 e ao Pedido de Patente U.K. No. GB1314263.3, intitulado "CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE" depositado em 8 de Agosto de 2013. Este pedido também reivindica o benefício da prioridade de data anterior do Pedido de Patente U.S. No. 14/458.052, intitulado "APPARATUS, SYSTEMS, AND METHODS FOR LOW POWER COMPUTATIONAL IMAGING" depositado em 12 de Agosto de 2014, que reivindica a prioridade ao Pedido de Patente Provisório No. 62/030.913, intitulado "LOW POWER COMPUTATIONAL IMAGING COMPUTING DEVICE" depositado em 30 de Julho de 2014 e ao Pedido de Patente U.S. No. 14/082.396, intitulado "APPARATUS, SYSTEMS, AND METHODS

FOR PROVIDING COMPUTATIONAL IMAGING PIPELINE” depositado em 18 de Novembro de 2013, que reivindica a prioridade do Pedido de Patente Romeno OSIM Registratura A/00812, intitulado “APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE” depositado em 6 de Novembro de 2013 e ao Pedido de Patente U.K. No. GB1314263.3, intitulado “CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE” depositado em 8 de Agosto de 2013. Cada um dos pedidos é aqui incorporado por referência neste documento em sua totalidade.

CAMPO DA INVENÇÃO

[0002] O presente pedido de patente refere-se de modo geral ao fornecimento um dispositivo de computação de imageamento computacional de baixo consumo de energia.

HISTÓRICO

[0003] O imageamento computacional é um novo paradigma da imageamento que é capaz de prover experiência de usuário sem precedentes e informações baseadas em imagens e vídeos. Por exemplo, imageamento computacional pode processar imagens e/ou vídeos para fornecer um mapa de profundidade de uma cena, fornecer uma vista panorâmica de uma cena, extrair faces de imagens e/ou vídeos, extrair texto, características e metadados de imagens e/ou vídeos e até mesmo fornecer capacidades de percepção visual automatizada baseada em objeto e características de reconhecimento de cena.

[0004] Enquanto o imageamento computacional pode fornecer capacidades interessantes, não foi amplamente

adotada. A lenta adoção de imageamento computacional pode ser atribuída ao fato de que o imageamento computacional vem com desafios de processamento de dados fundamentais. Frequentemente, resolução de imagem e taxas de quadro de vídeo são elevadas. Portanto, o imageamento computacional geralmente requer centenas de gigaflops de recursos computacionais, que podem ser difíceis de obter usando processadores de computador regulares, especialmente onde o desempenho tem de ser sustentável e apoiado pela elevada largura de banda de memória em dissipação de baixa potência. Além disso, o imageamento computacional é geralmente sensível à latência. Devido ao fato dos usuários não serem susceptíveis de esperar diversos minutos para uma câmera reconhecer um objeto, câmeras de imageamento computacional geralmente são projetadas para processar imagens e vídeos rapidamente, o que sobrecarrega a exigência computacional de imageamento computacional.

[0005] Infelizmente, é difícil de implementar técnicas de imageamento computacionais em hardware customizado. Visto que o campo de imageamento computacional está em sua relativa infância, técnicas de implementação estão em fluxo constante. Portanto, é difícil customizar imageamento computacional inteiramente em hardware conforme mudanças nas técnicas de implementação exigiriam novo projeto do hardware inteiro. Nesse sentido, é geralmente desejável fornecer uma arquitetura de hardware flexível e uma infraestrutura de hardware flexível.

[0006] Ao mesmo tempo, a demanda para tal processamento de vídeo e imagem está chegando a uma extensão grande de dispositivos eletrônicos portáteis, por exemplo computadores tablet e dispositivos móveis, onde o consumo de energia é uma consideração principal. Como resultado, há uma demanda geral por uma infraestrutura flexível de imageamento computacional que possa operar mesmo sob um montante restrito de potência.

RESUMO DA INVENÇÃO

[0007] De acordo com o objeto descrito, sistemas e métodos são providos para fornecer imageamento computacional de baixo consumo de energia.

[0008] O objeto descrito inclui um dispositivo de computação. O dispositivo de computação pode incluir uma pluralidade de processadores vetoriais, onde um processador, dentre a pluralidade de processadores vetoriais, é configurado para executar uma instrução que opera em uma primeira matriz de valores. O dispositivo de computação também pode incluir um acelerador de hardware configurado para realizar uma operação de filtragem em uma segunda matriz de valores. O dispositivo de computação também pode incluir um tecido de memória, compreendendo uma pluralidade de fatias de memória e um sistema de interconexão, tendo uma primeira interface e uma segunda interface, em que a primeira interface é configurada para acoplar a pluralidade de processadores vetoriais à pluralidade de fatias de memória e em que a segunda interface é configurada para acoplar o acelerador de hardware à pluralidade de fatias de memória. Ademais, o dispositivo de computação pode

incluir um processador host configurado para fazer com que o tecido de memória proveja a primeira matriz de valores para o processador dentre a pluralidade de processadores vetoriais através da primeira interface e para fornecer a segunda matriz de valores para o acelerador de hardware através da segunda interface, desse modo permitindo que o processador dentre a pluralidade de processadores vetoriais processe a primeira matriz de valores de acordo com a instrução e habilite o acelerador de hardware para processar a segunda matriz de valores de acordo com a operação de filtragem.

[0009] Em algumas modalidades, o dispositivo de computação pode incluir uma pluralidade de ilhas de energia, cada uma compreendendo pelo menos um domínio de energia, em que uma primeira dentre a pluralidade de ilhas de energia é acoplada a uma primeira voltagem de alimentação para fornecer a primeira voltagem de alimentação para um processador da pluralidade de processadores vetoriais, e em que uma segunda da pluralidade das ilhas de energia é acoplada a uma segunda voltagem de alimentação para fornecer a segunda voltagem de alimentação para o acelerador de hardware.

[0010] Em algumas modalidades, o dispositivo de computação pode incluir um módulo de gerenciamento de energia configurado para fornecer um sinal de habilitação para um comutador que acopla o primeiro da pluralidade das ilhas de energia para a primeira voltagem de alimentação, assim dispondo o processador da pluralidade de processadores vetoriais em um modo ativo.

[0011] Em algumas modalidades, o processador da pluralidade de processadores vetoriais pode compreender uma região de circuito lógico para processar a primeira matriz de valores e memória local para armazenar pelo menos um subconjunto da primeira matriz de valores, e em que o módulo de gerenciamento de energia pode ser configurado para fazer com que a primeira voltagem de alimentação seja fornecida para a região de circuito lógico e fazer com que uma terceira voltagem de alimentação seja fornecida para a memória local para controlar um consumo de energia da região de circuito lógico e da memória local de forma independente.

[0012] Em algumas modalidades, o módulo de gerenciamento de energia é configurado para desligar o comutador para desconectar a primeira da pluralidade de ilhas de energia a partir da primeira voltagem de alimentação, assim dispondo o processador da pluralidade de processadores vetoriais em um modo de baixo consumo de energia.

[0013] Em algumas modalidades, o módulo de gerenciamento de energia pode compreender um gerador de sinal válido configurado para gerar um sinal válido, indicando um momento no qual blocos de circuito na primeira da pluralidade de ilhas de energia estão prontos para processar os dados de entrada, em que o gerador de sinal válido compreende um encadeamento em série de comutadores que fornece a primeira voltagem de alimentação para os blocos de circuito na primeira da pluralidade de ilhas de energia.

[0014] Em algumas modalidades, o dispositivo de computação pode incluir um dispositivo periférico acoplado a uma pluralidade de pinos de entrada/saída (I/O), em que o dispositivo periférico é configurado para fornecer um canal de comunicação entre pelo menos um processador, dentre a pluralidade de processadores vetoriais e um dispositivo externo.

[0015] Em algumas modalidades, o dispositivo periférico pode estar dentro de uma ilha de energia que está sempre ligada.

[0016] Em algumas modalidades, o dispositivo periférico pode ser configurado para monitorar os sinais a partir do dispositivo externo para detectar um evento no qual um processador da pluralidade de processadores vetoriais deve responder, e quando o dispositivo periférico detectar o evento, fazer com que o módulo de gerenciamento de energia disponha o processador da pluralidade de processadores vetoriais no modo ativo.

[0017] Em algumas modalidades, o dispositivo periférico pode compreender um módulo de emulação que é configurado para fazer com que o dispositivo periférico emule uma funcionalidade de uma pluralidade de interfaces de protocolo padrão através de um conjunto comum de pinos de I/O.

[0018] Em algumas modalidades, o dispositivo periférico pode ser acoplado a um par diferencial de pinos de I/O, e o dispositivo periférico é configurado para mudar uma polaridade do par diferencial com base em um sinal de controle de polaridade.

[0019] Em algumas modalidades, o par diferencial de pinos de I/O pode compreender um par diferencial de pistas de interface de processador da indústria móvel (MIPI - Mobile Industry Processor Interface).

[0020] Em algumas modalidades, o dispositivo periférico pode compreender um buffer de bypass que é configurado para realizar um desvio entre um pino de I/O de entrada e um pino de I/O de saída, de modo a prover um canal de comunicação entre o pino de I/O de entrada e o pino de I/O de saída sem dispor o processador dentre os processadores vetoriais em um modo ativo.

[0021] O objeto descrito inclui um método. O método pode incluir prover um tecido de memória compreendendo uma pluralidade de fatias de memória e um sistema de interconexão tendo uma primeira interface e uma segunda interface. O método também pode incluir acoplar, usando a primeira interface, a pluralidade de fatias de memória e uma pluralidade de processadores vetoriais, e acoplar, usando a segunda interface, a pluralidade de fatias de memória e um acelerador de hardware. O método pode incluir adicionalmente prover, pelo tecido de memória, uma primeira matriz de valores para um processador, dentre a pluralidade de processadores vetoriais, através da primeira interface e prover uma segunda matriz de valores para o acelerador de hardware através da segunda interface, executar, em um dentre uma pluralidade de processadores vetoriais, uma instrução que opera na primeira matriz de valores, e realizar, através do acelerador de hardware, uma operação de filtragem na segunda matriz de valores.

[0022] Em algumas modalidades, o método pode incluir prover uma primeira voltagem de alimentação para um processador, dentre a pluralidade de processadores vetoriais, e prover uma segunda voltagem de alimentação para o acelerador de hardware, em que o processador dentre a pluralidade de processadores vetoriais e o acelerador de hardware estão associados a uma primeira ilha de energia e a uma segunda ilha de energia, respectivamente.

[0023] Em algumas modalidades, o método pode incluir prover, através de um módulo de gerenciamento de energia, um sinal de habilitação para um comutador que acopla a primeira ilha de energia à primeira voltagem de alimentação, assim dispondo o processador da pluralidade de processadores vetoriais em um modo ativo.

[0024] Em algumas modalidades, o método pode incluir gerar um sinal válido, indicando um momento no qual blocos de circuito na primeira ilha de energia estão prontos para processar dados de entrada, usando um encadeamento em série de comutadores que fornece a primeira voltagem de alimentação para os blocos de circuito no processador da pluralidade de processadores vetoriais.

[0025] Em algumas modalidades, o método pode incluir fornecer um dispositivo periférico acoplado a uma pluralidade de pinos de entrada/saída (I/O), no qual o dispositivo periférico é associado a uma ilha de energia que está sempre ligada.

[0026] Em algumas modalidades, o método pode incluir monitorar sinais a partir de um dispositivo externo para

detectar um evento no qual o processador da pluralidade de processadores vetoriais deve responder, e fazer com que o módulo de gerenciamento de energia disponha o processador da pluralidade de processadores vetoriais no modo ativo.

[0027] Em algumas modalidades, o método pode incluir emular, pelo dispositivo periférico, uma funcionalidade de uma pluralidade de interfaces de protocolo padrão através de um conjunto comum dos pinos de I/O.

[0028] Em algumas modalidades, o dispositivo periférico é acoplado a um par diferencial de pinos de I/O, e o método compreende adicionalmente mudar uma polaridade do par diferencial com base em um sinal de controle de polaridade.

[0029] Em algumas modalidades, o método pode incluir realizar um desvio entre um pino de I/O de entrada e um pino de I/O de saída usando um buffer de bypass, provendo assim um canal de comunicação entre o pino de I/O de entrada e o pino de I/O de saída sem dispor o processador, dentre os processadores vetoriais, em um modo ativo.

[0030] O objeto descrito inclui um dispositivo eletrônico. O dispositivo eletrônico pode incluir uma pluralidade de processadores vetoriais, em que um processador, dentre a pluralidade de processadores vetoriais, é configurado para executar uma instrução que opera em uma primeira matriz de valores. O dispositivo eletrônico também pode incluir um acelerador de hardware compreendendo um pipeline de percurso de dados programável que é programado usando informações de

configuração recebidas a partir de um módulo de software, em que o pipeline de percurso de dados programável é configurado para realizar uma operação de filtragem em uma segunda matriz de valores de acordo com as informações de configuração. O dispositivo eletrônico também pode incluir um tecido de memória compreendendo uma pluralidade de fatias de memória. O dispositivo eletrônico pode incluir adicionalmente um processador host configurado para fazer com que o tecido de memória proveja a primeira matriz de valores para o processador da pluralidade de processadores vetoriais e proveja a segunda matriz de valores para o acelerador de hardware, permitindo assim que o processador, dentre a pluralidade de processadores vetoriais, processe a primeira matriz de valores de acordo com a instrução e habilite o acelerador de hardware para processar a segunda matriz de valores de acordo com as informações de configuração.

[0031] Em algumas modalidades, o acelerador de hardware pode incluir um buffer de saída para receber uma linha de varredura de uma imagem processada pelo pipeline de percurso de dados programável, e um controlador de parada (*stall*) de pipeline configurado para parar uma operação do pipeline de percurso de dados programável quando o buffer de saída estiver cheio.

[0032] Em algumas modalidades, o acelerador de hardware pode incluir uma pluralidade de unidades funcionais que são encadeadas juntas para realizar a operação de filtragem.

[0033] Em algumas modalidades, uma ordem na qual a pluralidade de unidades funcionais é encadeada junta é

determinada usando as informações de configuração recebidas a partir do módulo de software.

[0034] Em algumas modalidades, uma saída de uma primeira da pluralidade de unidades funcionais é provida para uma buffer em um tecido de memória, e uma entrada de uma segunda da pluralidade de unidades funcionais é recebida a partir do buffer.

[0035] Em algumas modalidades, o acelerador de hardware pode incluir um cliente de mapa de profundidade que é configurado para receber informações de profundidade que são indicativas de uma profundidade de um objeto representado por um pixel na linha de varredura da imagem.

[0036] Em algumas modalidades, o acelerador de hardware pode incluir um módulo de mapa de profundidade que é configurado para processar as informações de profundidade para casar uma resolução das informações de profundidade com uma resolução da linha de varredura da imagem.

[0037] Em algumas modalidades, o módulo de mapa de profundidade é configurado para sincronizar temporalmente as informações de profundidade para a linha de varredura da imagem.

[0038] Em algumas modalidades, o tecido de memória inclui um controlador de mútua-exclusão (mutex) que é configurado para monitorar o status de uma requisição de acesso exclusivo, requisitando um acesso exclusivo a um recurso compartilhado por um dos processadores vetoriais, e quando o processador dentre os processadores vetoriais recebe um acesso exclusivo ao

recurso compartilhado, enviar uma mensagem de confirmação para o processador dentre os processadores vetoriais, indicando que o processador dentre os processadores vetoriais tem acesso exclusivo ao recurso compartilhado.

[0039] Em algumas modalidades, o tecido de memória pode incluir uma pluralidade de buffers, em que um primeiro da pluralidade de buffers é associado a um primeiro dos processadores vetoriais, e em que um segundo dos processadores vetoriais é configurado para enviar dados ao primeiro processador dentre os processadores vetoriais ao armazenar os dados na primeira dentre a pluralidade de buffers.

[0040] Em algumas modalidades, o tecido de memória é configurado para modificar dinamicamente uma capacidade do primeiro da pluralidade de buffers com base em uma quantidade de dados transferidos para o primeiro processador dentre os processadores vetoriais.

[0041] Em algumas modalidades, o tecido de memória é configurado para associar dinamicamente dois ou mais da pluralidade de buffers para o primeiro dos processadores vetoriais com base em uma quantidade de dados transferidos para o primeiro dos processadores vetoriais.

[0042] Em algumas modalidades, a pluralidade de buffers pode ser uma parte de uma dentre a pluralidade de fatias de memória no tecido de memória.

[0043] Em algumas modalidades, o tecido de memória pode ser configurado para armazenar as informações de estado de um dos processadores vetoriais, quando o

processador dentre os processadores vetoriais entra em um modo de baixo consumo de energia.

[0044] Em algumas modalidades, as informações de estado são armazenadas em uma memória de acesso aleatório estática no tecido de memória.

[0045] Em algumas modalidades, o tecido de memória pode incluir um controlador de acesso direto à memória (DMA), em que o controlador de DMA compreende uma lista de operações, indicando uma ordem na qual as operações de DMA devem ser realizadas.

[0046] Em algumas modalidades, o controlador de DMA pode ser configurado para realizar um subconjunto das operações de DMA na lista de operações com base em um buffer de habilitação, em que o buffer de habilitação inclui uma pluralidade de bits, em que um da pluralidade de bits é associado a uma das operações de DMA, e um valor do um da pluralidade de bits é indicativo de se a uma das operações de DMA deve ser realizada pelo controlador de DMA.

[0047] O objeto descrito inclui um método. O método pode incluir prover, através de um tecido de memória compreendendo uma pluralidade de fatias de memória, uma primeira matriz de valores para um de uma pluralidade de processadores vetoriais. O método também pode incluir prover, pelo tecido de memória, uma segunda matriz de valores para um acelerador de hardware compreendendo um pipeline de percurso de dados programável, executar, através de um processador, dentre a pluralidade de processadores vetoriais, uma instrução que opera na primeira matriz de valores, configurar o pipeline de

percurso de dados no acelerador de hardware usando informações de configuração, e realizar, usando o pipeline de percurso de dados no acelerador de hardware, uma operação de filtragem na segunda matriz de valores de acordo com as informações de configuração.

[0048] Em algumas modalidades, o método pode incluir receber, em um buffer de saída, uma linha de varredura de uma imagem processada pelo pipeline de percurso de dados programável; e parar (*stalling*), através de um controlador de parada de pipeline, uma operação do pipeline de percurso de dados programável quando o buffer de saída estiver cheio.

[0049] Em algumas modalidades, o acelerador de hardware compreende uma pluralidade de unidades funcionais, e o método inclui o encadeamento da pluralidade de unidades funcionais de acordo com as informações de configuração para realizar a operação de filtragem.

[0050] Em algumas modalidades, a pluralidade de unidades funcionais compreende uma primeira unidade funcional e uma segunda unidade funcional, e em que o encadeamento da pluralidade de unidades funcionais compreende uma saída da primeira unidade funcional para uma entrada da segunda unidade funcional.

[0051] Em algumas modalidades, o método pode incluir receber informações de profundidade que são indicativas de uma profundidade de um objeto representado por um pixel na linha de varredura da imagem; e sincronizar as informações de profundidade para a linha de varredura da imagem.

[0052] Em algumas modalidades, o método inclui monitorar, através de um controlador de memória no tecido de memória, um status de uma requisição de acesso exclusivo, requisitando um acesso exclusivo a um recurso compartilhado por um dos processadores vetoriais, e quando o processador, dentre os processadores vetoriais, recebe um acesso exclusivo ao recurso compartilhado, envia uma mensagem de confirmação a um dos processadores vetoriais, indicando que o processador, dentre os processadores vetoriais, tem o acesso exclusivo ao recurso compartilhado.

[0053] Em algumas modalidades, o tecido de memória pode incluir uma pluralidade de buffers, em que um primeiro da pluralidade de buffers é associado a um primeiro dos processadores vetoriais, e o método compreende adicionalmente o envio, através de um segundo processador, dentre os processadores vetoriais, de dados para o primeiro processador, dentre os processadores vetoriais, através do armazenamento dos dados na primeira da pluralidade de buffers.

[0054] Em algumas modalidades, o método pode incluir modificar dinamicamente uma capacidade do primeiro da pluralidade de buffers com base em uma quantidade de dados transferidos para o primeiro processador, dentre os processadores vetoriais.

[0055] Em algumas modalidades, o método pode incluir associar dinamicamente dois ou mais da pluralidade de buffers para o primeiro processador, dentre os processadores vetoriais, com base em uma quantidade de

dados transferidos para o primeiro processador, dentre os processadores vetoriais.

[0056] Em algumas modalidades, o método pode incluir armazenar informações de estado de um dos processadores vetoriais no tecido de memória quando o processador dentre os processadores vetoriais entra em um modo de baixo consumo de energia.

[0057] Em algumas modalidades, as informações de estado são armazenadas em uma memória de acesso aleatório estática no tecido de memória.

[0058] Em algumas modalidades, o método pode incluir manter, em um controlador de acesso direto à memória (DMA), uma lista de operações, indicando uma ordem na qual operações de DMA devem ser realizadas.

[0059] Em algumas modalidades, o método pode incluir realizar um subconjunto das operações de DMA na lista de operações com base em um buffer de habilitação, em que o buffer de habilitação inclui uma pluralidade de bits, em que um da pluralidade de bits é associado a uma das operações de DMA, e um valor do um da pluralidade de bits é indicativo de se a uma das operações de DMA deve ser realizada pelo controlador de DMA.

BREVE DESCRIÇÃO DOS DESENHOS

[0060] Os diversos objetos, características e vantagens do objeto descrito podem ser mais completamente apreciados com referência à seguinte descrição detalhada do objeto revelado quando considerado em conexão com os seguintes desenhos, em que numerais de referência similares identificam elementos similares. As figuras anexas são esquemáticas e não se

destinam a ser desenhadas em escala. Para fins de clareza, nem todo componente está identificado em cada figura. Nem todos os componentes de cada modalidade do objeto descrito são mostrados, onde a ilustração não é necessária para permitir que aqueles versados na técnica entendam o objeto descrito.

[0061] A Figura 1 provê uma ilustração de alto nível de um dispositivo de computação de acordo com algumas modalidades.

[0062] A Figura 2 mostra uma ilustração detalhada de um dispositivo de computação de acordo com algumas modalidades.

[0063] A Figura 3 ilustra um acelerador de hardware de acordo com algumas modalidades.

[0064] A Figura 4 ilustra um acelerador de hardware que pode se adaptar a uma operação de filtragem com base em informações de profundidade, de acordo com algumas modalidades.

[0065] A Figura 5 ilustra um acelerador de hardware.

[0066] A Figura 6 ilustra um acelerador de hardware baseado em funções genéricas de acordo com algumas modalidades.

[0067] A Figura 7 ilustra um acelerador de hardware que inclui um buffer de primeiro a entrar, primeiro a sair (FIFO) para comunicação entre módulos de função de processamento de sinais de imagem (ISP - Image Signal Processing) de acordo com algumas modalidades.

[0068] A Figura 8 ilustra uma fonte de alimentação chaveando (*gating*) uma ilha de energia de acordo com algumas modalidades.

[0069] A Figura 9 ilustra um gerador de sinal válido de acordo com algumas modalidades.

[0070] A Figura 10 ilustra um mecanismo de monitoramento de sinal de evento de acordo com algumas modalidades.

[0071] A Figura 11 mostra uma interface definida em software de acordo com algumas modalidades.

[0072] A Figura 12 mostra uma implementação detalhada de uma interface definida em software de acordo com algumas modalidades.

[0073] A Figura 13 ilustra um processador de eventos de acordo com algumas modalidades.

[0074] A Figura 14 ilustra um filtro de eventos em um processador de eventos de acordo com algumas modalidades.

[0075] A Figura 15 mostra um modo de bypass de um dispositivo periférico de acordo com algumas modalidades.

[0076] A Figura 16 mostra uma interface de processador da indústria móvel (MIPI) de acordo com algumas modalidades.

[0077] A Figura 17 ilustra uma aplicação de um mecanismo de inversão de polaridade para uma interface de entrada/saída de acordo com algumas modalidades.

[0078] A Figura 18 ilustra um tecido de memória tendo um controlador de exclusão mútua (mutex) baseado em hardware de acordo com algumas modalidades.

[0079] A Figura 19 ilustra uma atribuição dinâmica de buffers de acordo com algumas modalidades.

[0080] A Figura 20 ilustra um mecanismo de gerenciamento de energia que fornece diferentes voltagens para dispositivos de memória de circuitos lógicos de acordo com algumas modalidades.

[0081] A Figura 21 ilustra um mecanismo de acesso direto à memória (DMA) que implementa um mecanismo de habilitação de estrutura de dados de DMA baseado em buffer de acordo com algumas modalidades.

[0082] A Figura 22 ilustra um dispositivo eletrônico que inclui o dispositivo de computação de acordo com algumas modalidades.

DESCRIÇÃO DETALHADA

[0083] Na seguinte descrição, inúmeros detalhes específicos são estabelecidos em relação a sistemas e métodos do objeto descrito e o ambiente em que tais sistemas e métodos podem operar, etc., a fim de fornecer uma compreensão completa do objeto revelado. Entretanto, como será evidente para um versado na técnica, o objeto revelado pode ser praticado sem tais detalhes específicos, e que certas características, que são bem conhecidas na técnica, não são descritas em detalhes a fim de evitar a complicação do objeto revelado. Além disso, será entendido que os exemplos fornecidos abaixo são exemplares, e que é contemplado que existem outros sistemas e métodos que estão dentro do escopo do objeto revelado.

[0084] O imageamento computacional pode transformar a forma como as máquinas capturam e interagem com o mundo físico. Por exemplo, através de imageamento computacional, máquinas podem capturar imagens que eram

extremamente difíceis de capturar usando técnicas tradicionais de imageamento. Como outro exemplo, através de imageamento computacional, máquinas podem entender seu entorno e reagir de acordo com seu entorno.

[0085] Um dos desafios em trazer imageamento computacional para um mercado de massa é que o imageamento computacional é inerentemente dispendioso do ponto de vista computacional. O imageamento computacional utiliza frequentemente um grande número de imagens em uma alta resolução e/ou um grande número de vídeos com uma elevada taxa de quadros. Portanto, o imageamento computacional frequentemente necessita do suporte de plataformas computacionais poderosas. Ademais, devido ao fato do imageamento computacional ser frequentemente usado em configurações móveis, por exemplo, usando um smartphone ou um computador tablet, o imageamento computacional muitas vezes necessita do suporte de plataformas computacionais poderosas que podem operar com uma demanda de baixa potência.

[0086] A presente invenção descreve um dispositivo de computação que pode proporcionar uma plataforma computacional altamente capacitada, de baixo consumo de energia para imageamento computacional. A Figura 1 provê uma ilustração de alto nível de um dispositivo computacional de acordo com algumas modalidades. O dispositivo computacional 100 pode incluir uma ou mais unidades de processamento, por exemplo um ou mais processadores vetoriais 102 e um ou mais aceleradores de hardware 104, um tecido de memória inteligente 106, um

dispositivo periférico 108 e um módulo de gerenciamento de energia 110.

[0087] Os um ou mais processadores vetoriais 102 incluem uma unidade de processamento central (CPU) que implementa um conjunto de instruções que contém instruções que operam em um arranjo de dados chamado de vetores. Mais particularmente, os um ou mais processadores vetoriais 102 podem ser configurados para realizar operações aritméticas genéricas em um grande volume de dados simultaneamente. Em algumas modalidades, os um ou mais processadores vetoriais 102 podem incluir um processador de palavra de instrução muito longa, de múltiplos dados, única instrução (SIMD-VLIW). Em algumas modalidades, os um ou mais processadores vetoriais 102 podem ser projetados para executar instruções associadas a aplicativos de visão e imageamento por computador.

[0088] Os um ou mais aceleradores de hardware 104 incluem hardware de computador que realiza algumas funções mais rápido do que seria possível no software sendo executado em uma CPU de uso geral. Exemplos de um acelerador de hardware em aplicações que não sejam de visão incluem um módulo de aceleração de *blitting* em unidades de processamento de gráficos (GPUs) que é configurado para combinar diversos bitmaps em um, usando um operador *raster*.

[0089] Em algumas modalidades, os um ou mais aceleradores de hardware 104 podem fornecer uma infraestrutura configurável que é adaptada para aplicativos de processamento de imagens e visão por computador. Os aceleradores de hardware 104 podem ser

considerados para incluir hardware de empacotamento genérico para acelerar operações de processamento de imagens e visão por computador no entorno de um núcleo computacional de aplicação específica. Por exemplo, um acelerador de hardware 104 pode incluir um módulo de filtragem dedicado para a realização de operações de filtragem de imagem. O módulo de filtragem pode ser configurado para operar um kernel de filtro customizado em uma imagem de forma eficiente. Em algumas modalidades, o acelerador de hardware 104 pode emitir um pixel de saída totalmente computado por ciclo de clock.

[0090] O tecido de memória inteligente 106 pode ser configurado para fornecer um sistema de memória de baixa potência com pouca latência. Devido ao fato de que imagens e vídeos incluem uma grande quantidade de dados, é importante prover uma interface de alta velocidade entre a memória e as unidades de processamento. Em algumas modalidades, o tecido de memória inteligente 106 pode incluir, por exemplo, 64 blocos de memória, cada um dos quais inclui uma interface de 64 bits. Em tais modalidades, o tecido de memória 106 opera a 600MHz, por exemplo, é capaz de transferir dados a 307,2 GB/seg. Em outras modalidades, o tecido de memória inteligente 106 pode incluir qualquer outro número de blocos de memória, cada um dos quais pode incluir qualquer número de interfaces que implementam um ou mais protocolos de interface.

[0091] O dispositivo periférico 108 pode ser configurado para fornecer um canal de comunicação para envio e recepção de bits de dados para e a partir de

dispositivos externos, tal como um sensor de imagem e um acelerômetro. O dispositivo periférico 108 pode fornecer um mecanismo de comunicação para os processadores vetoriais 102, os aceleradores de hardware 104 e o tecido de memória 106 para se comunicar com dispositivos externos.

[0092] O módulo de gerenciamento de energia 110 pode ser configurado para controlar atividades de blocos designados dentro do dispositivo de computação 100. Mais particularmente, o módulo de gerenciamento de energia 110 pode ser configurado para controlar a voltagem de alimentação de blocos designados, também designados como ilhas de energia, dentro do dispositivo de computação 100. Por exemplo, quando o módulo de gerenciamento de energia 110 permite que uma fonte de alimentação de uma ilha de energia, o dispositivo de computação 100 pode ser acionado para fornecer uma voltagem de alimentação adequada para a ilha de energia. Em algumas modalidades, cada ilha de energia pode incluir um domínio de energia independente. Portanto, a fonte de alimentação das ilhas de energia pode ser controlada independentemente. Em algumas modalidades, o módulo de gerenciamento de energia 110 também pode ser configurado para controlar as atividades de ilhas de energia ligadas externamente ao dispositivo de computação 100 através de um ou mais dos pinos de entrada/saída do dispositivo de computação 100.

[0093] A Figura 2 apresenta uma ilustração detalhada de um dispositivo de computação de acordo com algumas modalidades. O dispositivo de computação 100 pode

incluir uma pluralidade de processadores vetoriais 102. Nesta ilustração, o dispositivo de computação 100 inclui 12 processadores vetoriais 102. Os processadores vetoriais 102 podem se comunicar entre si através da interconexão entre processadores (IPI) 202. Os processadores vetoriais 102 também podem se comunicar com outros componentes no dispositivo de computação 100, incluindo o tecido de memória 106 e/ou aceleradores de hardware 104, através do IPI 202 e da barra transversal (*crossbar*) de controlador de acelerador de memória (AMC) 204 ou um barramento de processador de mapeamento de memória 208.

[0094] Em algumas modalidades, os um ou mais processadores vetoriais 102 podem ser projetados para executar um conjunto de instruções proprietárias. O conjunto de instruções proprietárias pode incluir uma instrução proprietária. A instrução proprietária pode ser uma string binária de comprimento variável que inclui um cabeçalho de instrução e uma ou mais instruções de unidade. O cabeçalho de instruções pode incluir informações sobre o comprimento de instrução e as unidades ativas para a instrução proprietária associada; a instrução de unidade pode ser uma string binária de comprimento variável que inclui diversos campos que são fixos ou variáveis. Os campos na instrução de unidade podem incluir um opcode que identifica a instrução e um operando que especifica o valor de uso na execução da instrução de unidade.

[0095] Detalhes dos processadores vetoriais 102 são fornecidos no Pedido de Patente U.S. No. TBD, intitulado

"VECTOR PROCESSOR", identificado por um registro de advogado No. 2209599.127US1, depositado na mesma data do presente, que é aqui incorporada por referência em sua totalidade.

[0096] O dispositivo de computação 100 pode incluir aceleradores de hardware 104. Os aceleradores de hardware 104 podem incluir uma variedade de módulos aceleradores que são configurados para realizar funções de processamento predefinidas. Em algumas modalidades, uma função de processamento predefinidas pode incluir uma operação de filtragem. Por exemplo, os aceleradores de hardware 104 podem incluir um módulo de processamento de imagens brutas, um módulo de correção de sombreamento de lente (LSC), um módulo de *demosaicing* padrão bayer, um módulo de filtro de nitidez (*sharpen*), um módulo dimensionador polifásico, um módulo de detecção de canto de Harris, um módulo de combinação de cores, um módulo *denoise* de canal de luma, um módulo *denoise* de canal de chroma, um módulo de filtro mediano, uma tabela de consulta, um módulo de convolução, um módulo de detecção de borda e/ou qualquer outro módulo apropriado ou combinação de módulos. Os aceleradores de hardware 104 podem ser configurados para recuperar e armazenar dados em dispositivos de memória que residem no tecido de memória 106.

[0097] O tecido de memória 106 pode incluir um sistema de memória central que coordena as operações de memória dentro do dispositivo de computação 100. O tecido de memória 106 pode ser projetado para reduzir a transferência de dados desnecessária entre unidades de

processamento, tais como processadores vetoriais 102 e aceleradores de hardware 104. O tecido de memória 106 é construído de modo a permitir que uma pluralidade de unidades de processamento acessem, em paralelo, a memória de código de programa e dados sem parar. Ademais, o tecido de memória 106 pode prover que um processador *host* acesse o sistema de memória no tecido de memória 106 através de um barramento paralelo tal como a interface extensível avançada (AXI - Advanced eXtensible Interface) ou qualquer outro barramento 208 adequado.

[0098] Em algumas modalidades, uma unidade de processamento pode ler/gravar até 128 bits por ciclo através de suas portas de unidade de carga-armazenamento (LSU) e ler até 128 bits de código de programa por ciclo através de sua porta de instrução. Adicionalmente às interfaces IPI 202 e AMC 204 para processadores 102 e aceleradores de hardware 104, respectivamente, o tecido de memória 106 pode prover acesso simultâneo de leitura/gravação para um sistema de memória através do barramento de alto desempenho (AHB) de arquitetura avançada de barramento por microcontrolador (AMBA) e interfaces de barramento de AXI. O AHB e a AXI são barramentos de interface paralela padrão as quais permitem unidades de processamento, um sistema de memória e um dispositivo periférico para ser conectado usando uma infraestrutura de barramento compartilhada. Quaisquer outros barramentos adequados podem ser usados. Em algumas modalidades, o tecido de memória 106 pode ser configurado para lidar com um pico 18 x 128 bits de

acessos de memória por ciclo de clock. Em outras modalidades, o tecido de memória 106 pode ser projetado para lidar com qualquer número de acessos à memória por ciclo de clock, usando uma interface de alta velocidade com um grande número de bits.

[0099] Um sistema de memória no tecido de memória 106 pode incluir uma pluralidade de fatias de memória, cada fatia de memória sendo associada a um dos processadores vetoriais 102 e dando acesso preferencial àquele processador em relação a outros processadores vetoriais 102. Cada fatia de memória pode incluir uma pluralidade de recortes (*tiles*) de memória de acesso aleatório (RAM), onde cada recorte de RAM pode incluir uma porta de leitura e uma porta de gravação. Em alguns casos, cada fatia de memória pode ser fornecida com um controlador de fatia de memória para prover acesso a uma fatia de memória relacionada.

[00100] Os processadores e os recortes de RAM podem ser acoplados entre si através de um barramento, também referido como uma IPI 202. Em alguns casos, a IPI 202 pode acoplar qualquer um dos processadores vetoriais 202 a qualquer uma das fatias de memória no tecido de memória 106. Apropriadamente, cada recorte de RAM pode incluir um bloco de lógica de controle de recorte para conceder acesso ao recorte. O bloco de lógica de controle de recorte é muitas vezes referido como lógica de controle de recorte ou um bloco de arbitragem.

[00101] Em algumas modalidades, cada fatia de memória pode incluir uma pluralidade de recortes de RAM ou blocos de RAM físicos. Por exemplo, uma fatia de memória

tendo o tamanho de 128kB pode incluir quatro recortes de RAM de porta única de 32KB (por exemplo, elementos de RAM físicos) organizados como palavras de 4k x 32 bits. Como outro exemplo, uma fatia de memória tendo um tamanho de 256kB pode incluir oito recortes de RAM de porta única de 32KB (por exemplo, elementos de RAM físicos) dispostos como palavras de 8k x 32 bits. Em algumas modalidades, a fatia de memória pode ter uma capacidade tão baixa quanto 16kB e tão alta quanto 16MB. Em outras modalidades, a fatia de memória pode ser configurada para ter tanta capacidade quanto necessário para acomodar uma variedade de aplicativos gerenciados pelo dispositivo de computação.

[00102] Em algumas modalidades, um recorte de RAM pode incluir uma RAM semicondutora de metal-óxido complementar (CMOS) de porta única. A vantagem de uma CMOS RAM de porta única é que ela está geralmente disponível na maioria dos processos de semicondutores. Em outras modalidades, um recorte de RAM pode incluir uma RAM CMOS de múltiplas portas. Em algumas modalidades, cada recorte de RAM pode ter uma largura de 16 bits, largura de 32 bits, largura de 64 bits e largura de 128 bits, ou pode ser tão larga quanto necessário pelo aplicativo específico do dispositivo de computação.

[00103] O uso de dispositivos de memória de porta única pode aumentar a potência e eficiência de área do subsistema de memória, mas pode limitar a largura de banda do sistema de memória. Em algumas modalidades, o tecido de memória 106 pode ser projetado para permitir

que esses dispositivos de memória se comportem como um subsistema de memória de múltiplas portas virtual capaz de servir múltiplas requisições simultâneas de leitura e gravação a partir das múltiplas fontes (processadores e blocos de hardware). Isto pode ser conseguido usando vários casos de RAM físicos e fornecendo acesso arbitrado para eles para servir múltiplas fontes.

[00104] Em algumas modalidades, cada recorte de RAM pode ser associado à lógica de controle de recorte. A lógica de controle de recorte é configurada para receber requisições a partir de processadores vetoriais 102 ou aceleradores de hardware 104 e prover acesso às portas individuais de leitura e gravação do recorte de RAM associado. Por exemplo, quando um processador vetorial 102 está pronto para acessar dados em um recorte de RAM, antes do processador de vetor 102 enviar a solicitação de dados de memória para o recorte de RAM diretamente, o processador de vetor 102 pode enviar uma requisição de acesso à memória para a lógica de controle de recorte associada ao recorte de RAM. A requisição de acesso à memória pode incluir um endereço de memória de dados requisitados pelo elemento de processamento. Subsequentemente, a lógica de controle de recorte pode analisar a requisição de acesso à memória e determinar se o processador vetorial 102 pode acessar o recorte de RAM requisitado. Se o processador vetorial 102 puder acessar o recorte de RAM solicitado, a lógica de controle do recorte pode enviar uma mensagem de concessão de acesso para o processador vetorial 102, e subsequentemente, o processador vetorial 102 pode enviar

uma requisição de dados de memória para o recorte de RAM.

[00105] Em algumas modalidades, a lógica de controle de recorte pode ser configurada para determinar e impor uma ordem em que muitas unidades de processamento (por exemplo, processadores vetoriais e aceleradores de hardware) acessem o mesmo recorte de RAM. Por exemplo, a lógica de controle de recorte pode incluir um detector de confronto, que é configurado para detectar um caso no qual duas ou mais unidades de processamento tentam acessar um recorte de RAM simultaneamente. O detector de confronto pode ser configurado para reportar para um programador de tempo de execução que ocorreu um conflito de acesso e que o embate de acesso deve ser resolvido.

[00106] O tecido de memória 106 também pode incluir um barramento de memória para a transferência de bits de dados de memória para processadores vetoriais 102 ou aceleradores de hardware 104, ou de processadores vetoriais 102 ou aceleradores de hardware 104 para a memória. O tecido de memória 106 também pode incluir um controlador de acesso de memória direta (DMA) que coordena a transferência de dados entre processadores vetoriais 102, aceleradores de hardware 104 e memória.

[00107] Em algumas modalidades, os aceleradores de hardware 104 podem ser acoplados ao tecido de memória 106 através de um barramento separado. O barramento separado pode incluir um controlador de acelerador de memória (AMC) 204, que é configurado para receber requisições a partir do pelo menos um acelerador de hardware e a conceder, ao acelerador de hardware, um

acesso a uma fatia de memória através do controlador de fatia de memória relacionado. Assim, será apreciado que o percurso de acesso de memória empregado pelos aceleradores de hardware 104 pode ser diferente do percurso empregado pelos processadores vetoriais 102. Com efeito, o AMC 204 pode realizar filtragem de endereços, arbitragem e multiplexação. Em algumas modalidades, os aceleradores de hardware 104 podem incluir um buffer interno (por exemplo, uma memória FIFO) para compensar os retardos no acesso ao tecido de memória 106.

[00108] Em algumas modalidades, o AMC 204 pode ser acoplado a um ou mais dispositivos periféricos 108, incluindo, por exemplo, uma pluralidade de interfaces de câmera de interface de processador da indústria móvel (MIPI). O AMC 204 também pode ser conectado às interfaces AXI e APB para permitir que dois sistemas de processadores RISC acessem as fatias de memória no tecido de memória 106 através do AMC 204.

[00109] Em algumas modalidades, o AMC 205 pode incluir um par de portas de 64 bits em cada fatia de memória do tecido de memória 106. O AMC 204 pode ser configurado para rotear requisições a partir de um acelerador de hardware 104 para uma fatia de memória apropriada pelo decodificador de endereços parcial.

[00110] Em algumas modalidades, o AMC 204 pode ser acoplado a uma grande variedade de unidades de processamento para fornecer acesso a fatias de memória no tecido de memória 106. Por exemplo, o AMC 204 pode ser acoplado a qualquer tipo de aceleradores de hardware

ou elementos de terceiros para fornecer acesso a fatias de memória no tecido de memória 106. O AMC 204 também pode ser configurado para prover acesso a um espaço de memória maior de um sistema de computação, incluindo dispositivos de memória que se encontram fora do dispositivo de computação 100.

[00111] Em algumas modalidades, o AMC 204 pode arbitrar as requisições de acesso à memória simultaneamente para a mesma fatia de memória em sequência (*round-robin*). Por exemplo, uma unidade de processamento, tal como um acelerador de hardware 104, pode enviar, para o AMC 204, uma requisição de acesso à memória, que inclui um endereço de memória. Quando o AMC 204 receber a requisição de acesso à memória, o AMC 204 determina se o endereço de memória na requisição de acesso à memória está associado a uma fatia de memória no tecido de memória 106. Se o endereço de memória na requisição de acesso à memória não estiver associado com uma fatia de memória no tecido de memória 106, então, o AMC 204 pode encaminhar a requisição de memória ao mestre de AXI do AMC. Se o endereço de memória na requisição de acesso à memória estiver associado a uma fatia de memória no tecido de memória 106, o AMC 204 pode arbitrar a requisição de acesso à memória para fornecer acesso à posição de memória desejada.

[00112] O dispositivo periférico 108 pode ser configurado para fornecer um canal de comunicação para envio e recebimento de bits de dados para, e a partir de dispositivos externos, tais como sensores de imagem heterogêneas múltiplas e um acelerômetro. O dispositivo

periférico 108 pode fornecer um mecanismo de comunicação para os processadores vetoriais 102, os aceleradores de hardware 104 e o tecido de memória 106 para se comunicar com dispositivos externos.

[00113] Tradicionalmente, a funcionalidade de um dispositivo periférico foi corrigida e codificados (*hard-coded*). Por exemplo, periféricos da interface de processador da indústria móvel (MIPI) foram capazes apenas de interfacear com um dispositivo externo que também implementa interfaces digitais de baixa taxa, tais como a SPI, I2C, I2S ou quaisquer outras normas adequadas.

[00114] No entanto, em algumas modalidades da presente invenção, a funcionalidade do dispositivo periférico 108 pode ser definida usando software. Mais especificamente, o dispositivo periférico 108 pode incluir um módulo de emulação que é capaz de emular a funcionalidade de protocolos de interface padronizados, tais como a SPI, I2C, I2S ou qualquer outro protocolo adequado.

[00115] O módulo de gerenciamento de energia 110 é configurado para controlar as atividades de blocos dentro do dispositivo de computação 100. Mais particularmente, o módulo de gerenciamento de energia 110 é configurado para controlar a voltagem de alimentação de blocos designados, também conhecido como ilhas de energia. Por exemplo, quando o módulo de gerenciamento de energia 110 habilita uma fonte de alimentação de uma ilha de energia, o dispositivo de computação 100 é configurado para fornecer uma voltagem de alimentação adequada para a ilha de energia. O módulo

de gerenciamento de energia 110 pode ser configurado para habilitar uma fonte de alimentação de uma ilha de energia ao aplicar um sinal de habilitação em um registo ou em uma linha de sinal em um barramento. Em algumas modalidades, o módulo de gerenciamento de energia 110 também pode ser configurado para controlar atividades de dispositivo externo através de um ou mais pinos de entrada/saída do dispositivo de computação 100.

[00116] Em algumas modalidades, uma ilha de energia pode estar sempre ligada (por exemplo, a voltagem de alimentação é sempre fornecida para a ilha de energia). Uma tal ilha de energia pode ser referida como uma ilha de energia sempre ligada. Em algumas modalidades, a ilha de energia sempre ligada pode ser usada para monitorar sinais de, por exemplo, pinos de entrada e saída de uso geral (GPIO - General-Purpose-Input-Output), interfaces externas e/ou blocos funcionais internos tal como um temporizador de baixa frequência ou reset de inicialização. Desta forma, o dispositivo de computação 100 pode responder a um evento ou uma sequência de eventos e inicializar adaptativamente apenas ilhas de energia que são necessárias para responder ao evento ou à sequência de eventos.

[00117] A Figura 3 ilustra um acelerador de hardware de acordo com algumas modalidades. O acelerador de hardware 104 inclui uma coleção de filtros de processamento de imagens de hardware. O acelerador de hardware 104 pode habilitar algumas das funcionalidades intensivas em termos computacionais para que seja descarregado (*offloaded*) dos processadores vetoriais 102. O

acelerador 104 pode ser acoplado para o AMC 204 para acessar fatias de memória no tecido de memória 106 em uma elevada largura de banda.

[00118] Em algumas modalidades, o acelerador de hardware 104 pode ser acoplado ao tecido de memória 106 através do AMC 204. Em algumas modalidades, o acelerador de hardware 104 pode incluir um ou mais módulos de filtro (por exemplo, 20 módulos filtro), incluindo um filtro de receptor de MIPI e um filtro de transmissor de MIPI. Em algumas modalidades, um módulo de filtro pode incluir uma interface de AMC somente de leitura (uma interface de cliente de leitura) e uma interface de AMC apenas de gravação (uma interface de cliente de gravação). Em outras modalidades, um módulo de filtro também pode ter uma pluralidade de interfaces de AMC somente de leitura. Por exemplo, um módulo de filtro pode ter uma pluralidade de interfaces de AMC somente de leitura para um acesso paralelo para múltiplos buffers de entrada, vários planos (do mesmo buffer). A pluralidade de interfaces de AMC somente de leitura pode ser usada para fornecer que uma largura de banda de leitura de memória extra para manter a vazão de processamento do módulo de filtro. As descrições de um acelerador de hardware 104 podem ser igualmente aplicáveis para cada módulo de filtro, desde que um acelerador de hardware 104 possa apenas ter um único módulo de filtro. Da mesma forma, as descrições de um módulo de filtro podem ser igualmente aplicáveis a um acelerador de hardware desde que o módulo de filtro

possa ser o único módulo de filtro no acelerador de hardware.

[00119] Em algumas modalidades, o AMC 204 apresenta uma ou mais portas bidirecionais (por exemplo, leitura/gravação) em cada fatia de memória no tecido de memória 106. As portas podem acomodar um grande número de bits. Por exemplo, as portas podem acomodar uma comunicação de 64 bits. Em algumas modalidades, o AMC 204 também pode incluir um mestre de AXI, que provê uma conectividade direta com dispositivos de DRAM externos.

[00120] Em algumas modalidades, um módulo de filtro pode ser projetado principalmente para processar buffers no tecido de memória 106. Por exemplo, com exceção de um módulo receptor de MIPI e um módulo de filtro de transmissão de MIPI, um módulo de filtro pode alimentar e obter de dados apenas através de seus clientes AMC. A configuração dos módulos de filtro, incluindo seus endereços de base de buffer, pode ser alcançada através de diversas interfaces escravo de APB.

[00121] Em algumas modalidades, o acelerador de hardware 104 pode receber dados de imagem através de um módulo de filtro de receptor de MIPI e um módulo de filtro de transmissor de MIPI. O módulo de filtro de receptor de MIPI e o módulo de filtro de transmissor de MIPI podem permitir que outros módulos de filtro no acelerador de hardware 104 estabeleça uma conexão direta com um controlador de receptor de MIPI e um controlador de transceptor de MIPI. O módulo de filtro de receptor de MIPI e o módulo de filtro de transmissor de MIPI podem se conectar a controladores de MIPI através de

interfaces paralelas e podem ser usados para transmitir dados em fluxo contínuo para dentro/para fora do tecido de memória 106 diretamente de/para o controlador de MIPI.

[00122] Em algumas modalidades, o acelerador de hardware 106 pode operar em linhas de varredura de dados de imagem armazenados em buffer no tecido de memória 106, acessado através do AMC 204. O AMC 204 pode rotear transações de suas interfaces de cliente para a fatia de memória de destino (ou o mestre de AXI) e arbitrar entre transações simultâneas de diferentes clientes em cada fatia de memória. Em algumas modalidades, os módulos de múltiplos filtro no acelerador de hardware 106 podem ser conectados juntos em uma maneira de fluxo contínuo por acoplamento de um buffer de saída de um ou mais módulos de filtro (também referidos como produtores/pais) para buffers de entrada de outros módulos de filtro (também referidos como consumidores/crianças).

[00123] Em algumas modalidades, um módulo de filtro em um acelerador de hardware 104 pode operar kernel bidimensional em pixels centrados no pixel atual. Todos os pixels no kernel podem contribuir no processamento de pixels centrados no pixel atual.

[00124] Em algumas modalidades, um módulo de filtro em um acelerador de hardware 104 pode processar uma imagem linha por linha. Por exemplo, um módulo de filtro pode varrer uma imagem de cima para baixo para gerar uma linha de varredura de uma imagem e processar as linhas de varredura, por exemplo, movendo-se da esquerda para a direita. Em outros exemplos, um módulo de filtro pode

gerar linhas de varredura de uma imagem por varredura de uma imagem em qualquer orientação/ordenação apropriada para o processamento de filtro.

[00125] Em algumas modalidades, um módulo de filtro pode processar uma linha de varredura de uma imagem através da leitura de dados para formar um kernel para um primeiro pixel na linha de varredura. O módulo de filtro pode processar a linha de varredura ao deslizar o kernel em forma de janela deslizante. Quando o processamento estiver concluído, o módulo de filtro pode gravar os pixels de saída em um buffer de saída ou um local de memória.

[00126] Em algumas modalidades, os kernels para filtragem são geralmente quadrados em muitas vezes têm um número ímpar de pixels ao longo de cada lado, por exemplo, 3x3, 5x5 ou 7x7. Se um módulo de filtro usar um kernel de pixel $K \times K$ e, em seguida, K -linhas de varredura de dados de imagem podem ser lidas em um buffer de entrada para cada linha de dados de imagem processada e gravada para o seu buffer de saída.

[00127] Em algumas modalidades, o acelerador de hardware 104 pode usar um buffer de entrada circular. Suponha que um módulo de filtro de destino esteja configurado para receber, como entrada, uma linha de varredura de saída de outro módulo de filtro (também referido como um módulo de filtro pai). Suponha também que o módulo de filtro de destino usa um kernel de pixel $K \times K$. Em seguida, o buffer de entrada para o módulo de filtro de destino pode ser projetado para manter pelo menos $(K+1)$ linhas de varredura de dados de imagem: K

linhas de varredura para o módulo de filtro e uma (ou mais) linha de varredura para receber simultaneamente uma linha de varredura de saída do módulo de filtro pai. Neste exemplo, devido ao fato do buffer de entrada ser circular, depois de receber $(K+1)$ linhas de varredura a partir do módulo de filtro pai, a $(K+2)$ -ésima linha de varredura pode ser gravada sobre o local da primeira linha. Na maioria dos casos, o módulo de filtro pai pode estar à frente do módulo de filtro de destino em termos de seu número de linha atual dentro da imagem de entrada. Após a configuração inicial, os clientes de AMC de leitura e gravação dos módulos de filtro podem cuidar do empacotamento de endereço de buffer circular ao acessar os buffers de entrada e saída dos módulos de filtro.

[00128] Em algumas modalidades, os buffers no acelerador de hardware 104 podem ser alinhados por um número predeterminado de bytes. Por exemplo, os buffers no acelerador de hardware 104 podem ser alinhados em limites de 8 bytes. Para facilitar o roteamento de transações, os clientes de leitura e gravação e o AMC podem ser configurados para prover apenas acessos de buffer alinhados. Quando uma largura da imagem não for um múltiplo de um número predeterminado de bytes, o acelerador de hardware 104 pode ser configurado para gravar bytes nulos nos buffers de saída entre o final (não alinhado) de cada linha de varredura e o próximo limite de byte.

[00129] A Figura 3 ilustra uma implementação de um acelerador de hardware para operar um kernel de filtro,

armazenar em um registro de kernel de filtro 302, em um fluxo de dados de entrada (por exemplo, linhas de varredura de uma ou mais imagens). Os fluxos de dados de entrada podem corresponder a pixels em uma ou mais imagens. O acelerador de hardware 104 pode incluir um pipeline de percurso de dados 304, um controlador de parada de pipeline 306, um cliente de leitura de buffer de linha 308, uma entrada de controle de início de linha 310 e um cliente de gravação de buffer de linha 310. Em algumas modalidades, um acelerador de hardware 104 pode incluir pelo menos uma interface de cliente de leitura de AMC 314 e/ou pelo menos uma interface de cliente de gravação de AMC 316 para acessar uma fatia de memória no tecido de memória 106. O número de interfaces de cliente de leitura/gravação no AMC 204 é adequadamente configurável.

[00130] Em algumas modalidades, o registrador de kernel de filtro 302 pode ser programado para modificar o kernel a ser operado no fluxo de dados de entrada. O registrador de kernel de filtro 302 pode ser configurado para acomodar uma variedade de tamanhos de kernel. Por exemplo, o registrador de kernel de filtro 302 pode ser configurado para acomodar um kernel 3x3, um kernel 5x5, um kernel 7x7, um kernel 9x9 ou quaisquer outros tamanhos de kernel representados como $m \times n$. Em alguns casos, m pode ser o mesmo que n ; em outros casos, m pode ser diferente de n . Em algumas modalidades, o registrador de kernel de filtro 302 pode ser configurado para acomodar kernels de várias dimensões. Por exemplo, o registrador de kernel de filtro 302 pode ser

configurado para acomodar um filtro unidimensional, um filtro bidimensional, um filtro tridimensional ou quaisquer filtros de número de filtros de número inteiro.

[00131] Em algumas modalidades, o cliente de leitura de buffer de linha 308 é configurado para receber uma linha de varredura de uma imagem (por exemplo, uma linha ou uma coluna de uma imagem em uma grade de imagem) e para prover a linha de varredura para o pipeline de percurso de dados 304. O cliente de leitura de buffer de linha 308 pode receber a linha de varredura de uma imagem a partir de uma interface de leitura de AMC 314. Uma vez que o pipeline de percurso de dados 304 recebe um kernel e uma linha de varredura de uma imagem, o pipeline de percurso de dados 304 pode realizar a operação de filtragem. Uma vez que o pipeline de percurso de dados 304 completa a operação de filtragem, o pipeline de percurso de dados 304 pode armazenar a linha resultante no cliente de gravação de buffer de linha 312. O cliente de gravação de buffer de linha 312 pode, opcionalmente, armazenar a linha resultante em uma fatia de memória através de uma interface de gravação de AMC 316. O controlador de parada de pipeline 306 pode parar determinadas partes do pipeline para assegurar que o cliente de gravação de buffer de linha 312 não transborde (*overflow*).

[00132] Em algumas modalidades, o controlador de início de linha 310 pode controlar um momento no qual o pipeline de percurso de dados 304 começa a processar a linha de varredura recebida de uma imagem. O controlador

de início de linha 310 também pode ser configurado para habilitar seletivamente uma ou mais partes do pipeline de percurso de dados 304 para realizar operações customizadas. Em alguns casos, o controlador de início de linha 310 também pode controlar os coeficientes a ser usados durante a operação de filtragem pelo pipeline de percurso de dados 304.

[00133] Em algumas modalidades, o pipeline de percurso de dados 304 e o controlador de início de linha 310 também podem ser programáveis. O pipeline de percurso de dados 304 e o controlador de início de linha 310 podem ser programados para que diferentes tipos de operações de filtragem possam ser realizadas pelo acelerador de hardware 104. Por exemplo, o pipeline de percurso de dados 304 e o controlador de início de linha 310 podem ser programados com parâmetros de operação do filtro, tais como os ajustes de coeficiente e/ou limiares, para que a operação de filtragem customizada possa ser realizada pelo acelerador de hardware 104. Os parâmetros de operação do filtro também podem incluir um tamanho de kernel de filtro, coeficientes, taxas de escalonamento, ganhos, limiares, tabelas de consulta ou quaisquer outros parâmetros adequados. Portanto, o acelerador de hardware 104 pode ser considerado como um empacotamento genérico para acomodar várias operações de filtragem de imagem.

[00134] Em algumas modalidades, o pipeline de percurso de dados 304 pode ser configurado para processar números representados em um ou mais formatos de número. Por exemplo, o pipeline de percurso de dados 304 pode ser

projetado para operar em números de ponto flutuante, por exemplo, fp16 (formato de ponto flutuante de 16 bits tipo IEEE754), números inteiros, números de ponto fixo ou quaisquer outros formatos de número adequados para processamento de imagem.

[00135] O acelerador de hardware 104 pode ser configurado para controlar como o pipeline de percurso de dados 304 consome linhas de varredura a partir de um buffer de dados de entrada 308 e como o pipeline de percurso de dados 304 armazena linhas de varredura processadas para um buffer de dados de saída 312. O acelerador de hardware 104 pode ser configurado para implementar um dentre os dois modos de controle: o modo de controle de preenchimento de buffer (BFC - Buffer Fill Control) e o modo síncrono.

[00136] Em algumas modalidades, sob a modalidade do BFC, o acelerador de hardware 104 pode ser configurado para manter contagens internas dos níveis de preenchimento (por exemplo, o número de linhas de varredura armazenadas no buffer de entrada). O acelerador de hardware 104 pode ser configurado para processar uma linha de varredura de seu buffer de entrada autonomamente quando (1) o acelerador de hardware estiver habilitado, (2) o buffer de entrada tem um número suficiente de linhas de varredura e (3) há espaço no seu buffer de saída para armazenar uma linha de varredura processada. Em alguns casos, o nível de preenchimento de buffer necessário para executar o pipeline de percurso de dados 304 pode depender do nível (*height*) de um kernel. Por exemplo, quando um kernel for

3x3, o acelerador de hardware 104 pode requerer pelo menos três linhas de varredura para operar um filtro.

[00137] Em algumas modalidades, sob um modo de controle síncrono, um módulo de filtro em um acelerador de hardware pode ser configurado para executar quando um bit de início para o módulo de filtro estiver ativado. O bit de início pode ser ligado usando, por exemplo, um módulo de software. Sob controle síncrono, o módulo de software pode ser configurado para determinar que o buffer de entrada para o módulo de filtro possui um número suficiente de linhas de varredura e que o buffer de saída para o módulo de filtro tem espaço suficiente para armazenar linhas de varredura processadas do módulo de filtro. Uma vez que estas condições são satisfeitas, o módulo de software pode ativar o bit de início do módulo de filtro.

[00138] Em ambos os modos, uma vez que um módulo de filtro processa uma linha de varredura, o módulo de filtro pode atualizar seu índice de linha atual dentro de seu buffer e dentro da imagem de entrada. Em algumas modalidades, quando a imagem de saída não tem o mesmo tamanho que a imagem de entrada, o módulo de filtro também pode atualizar seu índice de linha atual da imagem de saída. Os valores dos índices da linha (e níveis de preenchimento de buffer para controle de preenchimento de buffer) podem representar o estado interno de um módulo de filtro. Esse estado interno pode ser acessado por um módulo de software e pode ser salvo, atualizado e restaurado de tal forma que o contexto do

módulo de filtro possa ser comutado antes do módulo de filtro ser executado no próximo ciclo.

[00139] Em algumas modalidades, os buffers em um acelerador de hardware 104 podem ser configurados para manter uma pluralidade de planos de dados. Por exemplo, os buffers em um acelerador de hardware 104 podem ser configurados para manter o canal vermelho, o canal verde e o azul-canal de uma imagem em planos separados. Em alguns exemplos, os buffers em um acelerador de hardware 104 podem ser configurados para suportar até dezesseis planos. As linhas de varredura de dados de imagem em cada plano podem ser armazenadas de forma contígua e os planos podem ser definidos pelo seu número e por um passo de plano (*plane stride*).

[00140] Em algumas modalidades, um módulo de filtro em um acelerador de hardware 104 pode ser configurado para processar uma linha de varredura de cada plano de dados sequencialmente, um de cada vez. Para processamento sequencial, a partir do ponto de vista de controle, linhas de varredura de todos os planos podem ser considerados como tendo o mesmo registro temporal. Em outras modalidades, um módulo de filtro em um acelerador de hardware 104 pode ser configurado para processar múltiplos planos de dados em paralelo.

[00141] Em algumas modalidades, antes de processar um fluxo de imagem/vídeo, ou se o contexto for comutado, um módulo de filtro pode ser apropriadamente configurado e habilitado. Cada módulo de filtro pode incluir um conjunto de registros programável de software definindo

seus buffers de entrada e a configuração de buffer de saída.

[00142] Em algumas modalidades, um buffer em um módulo de filtro pode ser programado usando um ou mais dos seguintes parâmetros:

- **base:** Endereço de base. Este parâmetro pode especificar o endereço de base do buffer.

Endereços podem ser alinhados em limites de byte (por exemplo, a largura do barramento de dados de cliente de AMC).

- **nl:** Número de linhas de varredura. No modo de buffer circular, este parâmetro pode especificar o tamanho de um buffer circular em linhas de varredura. O número máximo de linhas de varredura para um buffer circular pode ser 1023, mas outros limites superiores também são possíveis. Se um buffer for configurado com $nl = 0$, isso indica que o buffer está em um modo não circular. Portanto, $nl = 0$ coloca o acesso de cliente(s) de leitura/gravação ao buffer em um modo não circular ou de não empacotamento em que o número de linhas de varredura no buffer corresponde ao nível da imagem e nenhum empacotamento de ponteiro de buffer circular ocorre.

- **Is:** Passo de linha. O passo da linha pode ser um múltiplo de um número fixo de bytes, por exemplo, 8 bytes. O passo de linha máximo pode ser predeterminado. O passo de linha máximo pode ser de (32MB - 8) bytes, por exemplo. O passo da linha e o número de linhas podem ser usados por clientes de leitura/gravação para realizar a aritmética de ponteiro de buffer circular. O

passo da linha pode ser maior ou igual à largura da imagem.

- **np:** Número de planos. Este parâmetro indica um número de planos, representado por um buffer. Quando $np = 0$, indica que o buffer representa dados não planos (por exemplo, um único plano de dados). A quantidade de armazenamento em buffer de linha em um buffer pode ser multiplicada pelo número de planos.

- **ps:** Passo de plano. O passo de plano pode ser um múltiplo de um número fixo de bytes, por exemplo, 8 bytes. O passo de plano máximo pode ser predeterminado. Por exemplo, o passo de plano máximo pode ser de (32MB - 8) bytes. Normalmente, o passo de plano pode ser maior que ou igual a nl multiplicado por ls . No entanto, outro passo de plano pode ser possível.

- **formato:** Formato de dados de buffer. Este parâmetro pode especificar o tamanho dos dados de pixel em bytes. Por exemplo, para um buffer de FP16, o formato pode ser definido como 2, indicando 2 bytes por pixel.

[00143] Em algumas modalidades, um buffer de saída em um módulo de filtro pode ser programado usando um ou mais dos seguintes parâmetros:

- **deslocamento:** O deslocamento pode especificar o deslocamento do endereço de base (e o início de cada linha) para o primeiro pixel. Este parâmetro pode ser usado para contornar a limitação de buffers sendo alinhados em um limite de byte. Usando o deslocamento, um espaço pode ser reservado à esquerda das linhas de varredura, por exemplo para enchimento de pixel horizontal pelo consumidor de um buffer de saída. O

padrão de deslocamento é zero. Se for especificado um deslocamento diferente de zero, então os bytes nulos podem ser gravados em cada linha de varredura de saída antes do primeiro pixel de saída.

[00144] Em algumas modalidades, um módulo de filtro pode suportar a uma variedade de tipos de dados. Os tipos de dados mais comuns suportados por um módulo de filtro estão listados abaixo:

- U8 - dados de número inteiro de 8 bits não designados
- U8F - dados fracionários de 8 bits na faixa [0, 1,0] não designados
- U16 - dados de número inteiro de 16 bits não designados
- U32 - dados de número inteiro de 32 bits não designados
- FP16 - ponto flutuante com metade de precisão (16 bits)
- FP32 - ponto flutuante com precisão completa (32 bits)

[00145] Em algumas modalidades, o pipeline de percurso de dados de um módulo de filtro pode ser otimizado para seu funcionamento: aritmética de ponto flutuante com metade de precisão (FP16) pode ser usada para operações envolvendo uma alta faixa dinâmica; aritmética de ponto fixo otimizada pode ser usada onde a manutenção de alta precisão é mais importante.

[00146] Em algumas modalidades, um módulo de filtro implementado usando uma aritmética FP16 não pode ser restrita à leitura/gravação somente aos buffers FP16. Os

buffers U8F também podem ser acessados com conversão de/para FP16 tomando lugar automaticamente dentro dos módulos de filtro.

[00147] Em algumas modalidades, onde um módulo de filtro é implementado usando aritmética de FP16, os buffers podem ser FP16 ou U8F. Quando uma buffer for FP16, o formato de configuração de buffer pode ser ajustado como 2. Se um buffer for U8F, o formato de configuração de buffer pode ser ajustado como 1. Para módulos de filtro com pipeline de percurso de dados FP16, se o formato de buffer de entrada for "1", o cliente de leitura pode converter os dados de entrada U8F para FP16 automaticamente antes do processamento. Se o formato de buffer de saída for "1", o cliente de gravação pode converter FP16 do pipeline de percurso de dados para U8F antes do armazenamento.

[00148] Em algumas modalidades, U8F é convertida em FP16 normalizado, na faixa $[0, 1, 0]$, multiplicando-se por $1,0/255$. FP16 normalizado pode ser convertido em U8F multiplicando-se por 255 e arredondando-se, quantizando efetivamente os valores de ponto flutuante em 8 bits. Em algumas modalidades, os dados de saída de módulos de filtro com pipeline de percurso de dados de FP16 pode, opcionalmente, ser fixado na faixa normalizada $[0, 1, 0]$. Se a conversão para U8F for habilitada, a fixação para a faixa normalizada é habilitada implicitamente e é executado antes da conversão para U8F descrita acima. Módulos de filtro implementados usando pipelines de percurso de dados FP16 não estão limitados ao

processamento de dados na faixa normalizada $[0, 1, 0]$; a faixa completa de FP16 também pode ser suportada.

[00149] Algumas modalidades, um módulo de filtro é configurado para rastrear sua posição vertical em uma imagem de entrada. Um módulo de filtro pode usar estas informações para executar o enchimento vertical na parte superior e inferior da imagem por replicação de linha ou reflexão. Um módulo de filtro que não realiza enchimento vertical pode criar uma imagem de saída que é menor do que uma imagem de entrada, o que pode não ser desejável em alguns casos.

[00150] Em algumas modalidades, quando um módulo de filtro é configurado para executar o enchimento vertical, o número mínimo de linhas de varredura M que pode ser mantido por um buffer de entrada pode ser:

$M = (K \gg 1) + 1$, onde \gg indica um operador de deslocamento de bits para direita.

[00151] Na parte superior da imagem, quando a capacidade do buffer de entrada (em termos de linhas de varredura) for menor que M , não há linhas de varredura suficientes no buffer para realizar a operação de filtragem. Quando a capacidade do buffer de entrada (em termos de linhas de varredura) for maior que ou igual a M , os dados podem ser processados se o enchimento vertical for realizado. Da mesma forma, na parte inferior da imagem, ao se processar as últimas linhas $(K \gg 1)$, o módulo de filtro pode realizar a replicação da linha $N-1$ (ou reflexão da linha $N-1$ e das linhas acima dela).

[00152] Em algumas modalidades, o enchimento vertical pode ser realizado quando o kernel tem uma dimensão par.

O enchimento vertical para um kernel com uma dimensão par pode ser virtualmente idêntico ao enchimento vertical para um kernel com uma dimensão ímpar, exceto pelo fato de que uma linha a menos deve ser preenchida na parte inferior.

[00153] Em algumas modalidades, um módulo de filtro pode realizar um enchimento horizontal. O enchimento horizontal de um kernel de pixel pode ser realizado conforme os dados são lidos do buffer de entrada e gravados nos registradores de kernel de pixel. O módulo de filtro pode estar ciente de sua posição na linha atual e no início e no final de uma linha. Portanto, registradores de kernel de pixel válido podem ser replicados para aqueles que não possuem dados válidos. Tal como com o enchimento vertical, se o enchimento horizontal for realizado ou não, pode depender da funcionalidade específica e requisitos de um módulo de filtro determinado.

[00154] Em algumas modalidades, em um modo de buffer circular, um módulo de filtro pode ser configurado para processar uma linha de varredura de seu buffer de entrada e gravar a linha de varredura processada para seu buffer de saída. Este conjunto de operações pode ser referido como uma execução de filtro.

[00155] Em algumas modalidades, para flexibilidade, dois mecanismos de controle diferentes podem ser providos pelo qual as execuções de filtro possam ser controladas. No primeiro mecanismo, chamado de modo de controle de preenchimento de buffer, um módulo de filtro pode rastrear os níveis de preenchimento de seus buffers

circulares e determinar, por conta própria, se pode ser executado. Esta abordagem é assíncrona por natureza; o módulo de filtro pode executar, possivelmente repetidamente, desde que sejam satisfeitas as condições requeridas. Bits de controle nos registradores são fornecidos para permitir que o software informe os módulos de filtro quando uma linha de varredura foi adicionada a um buffer de entrada ou removida de um buffer de saída. Quando uma linha de varredura for adicionada a um buffer de entrada, o nível de preenchimento pode ser aumentado; quando uma linha de varredura for removida de um buffer de saída, o nível de preenchimento pode ser reduzido. Deste modo, um módulo de filtro, juntamente com seus buffers de entrada e de saída, podem ser vistos como buffers de primeiro a entrar, primeiro a sair (FIFO) com linhas de varredura ocupando as suas entradas e a profundidade do FIFO configurada pelo número de linhas de varredura programado para os buffers de entrada e saída.

[00156] Em algumas modalidades, outro módulo de filtro pode adicionar uma linha de varredura para o FIFO se o buffer de entrada do módulo de filtro não estiver cheio. O software pode verificar o nível de preenchimento de um buffer de entrada antes de permitir que o outro módulo de filtro adicione uma linha de varredura para o buffer de entrada. Subsequentemente, o software ou um módulo de filtro pode aumentar um nível de preenchimento associado ao buffer de entrada. No lado de saída, o software pode verificar o nível de preenchimento do buffer de saída, ou responder a um

evento de interrupção, significando que um módulo de filtro adicionou uma nova linha de varredura para o buffer de saída, antes de decrementar o nível de preenchimento do buffer de saída (por exemplo, depois que uma linha no buffer de saída do filtro foi processada por outro filtro, tal como lendo o FIFO).

[00157] O segundo mecanismo, designado de um modo síncrono, depende do software para programar explicitamente cada execução de filtro. Bits de início para cada módulo de filtro podem ser fornecidos em registradores para os quais software pode gravar para iniciar a execução de filtro imediatamente. Quando iniciado por este mecanismo, um módulo de filtro pode ser executado exatamente uma vez.

[00158] Em algumas modalidades, um módulo de filtro pode ser interrompido quando ele recebe uma requisição de interrupção. Em alguns casos, um módulo de filtro pode ter uma pluralidade de fontes de requisição de interrupção que são mapeados para as linhas de requisição de interrupção externa e roteadas a um controlador de interrupção. Quando um módulo de filtro sinaliza uma interrupção e essa interrupção é habilitada, em seguida a linha de requisição de interrupção externa correspondente pode ser sinalizada (*flagged*).

[00159] Em algumas modalidades, a pluralidade de fontes de requisição de interrupção pode incluir:

- Interrupção de decremento de nível de preenchimento de buffer de entrada

- Interrupção de decremento de nível de preenchimento de buffer de saída

- Interrupção de quadro concluído

[00160] A interrupção de incremento de nível de preenchimento de buffer de saída também pode ser considerada para indicar que um módulo de filtro terminou sua execução de filtro quando o módulo de filtro for configurado para operar no modo síncrono.

[00161] Em algumas modalidades, o acelerador de hardware 104 pode adaptar-se à operação de filtragem com base em informações de profundidade. Por exemplo, o acelerador de hardware 104 pode ser configurado para borrar condicionalmente apenas os pixels associados a objetos que estejam a mais do que 27,4 metros (30 jardas) de distância ou pixels que estejam além de 4,5 metros (5 jardas) poderiam ser borrada menos do que aqueles que estejam além de 9,1 metros (10 jardas), etc.

[00162] A Figura 4 ilustra um acelerador de hardware que pode se adaptar a uma operação de filtragem com base em informações de profundidade de acordo com algumas modalidades. O acelerador de hardware ciente de profundidade 402 inclui, além de módulos no acelerador de hardware 104 na Figura 3, um cliente de leitura de mapa de profundidade 404 e um módulo de mapa de profundidade 406. O cliente de leitura de mapa de profundidade 404 é configurado para receber um mapa de profundidade que indica uma profundidade de um objeto representado por um pixel na imagem correspondente. Por exemplo, quando o cliente de leitura de buffer de linha 308 recebe uma linha de varredura de uma imagem, o

cliente de leitura de mapa de profundidade 404 pode ser configurado para receber um mapa de profundidade correspondente à linha de varredura da imagem.

[00163] Subsequentemente, o cliente de leitura de mapa de profundidade 404 pode fornecer o mapa de profundidade para o módulo de mapa de profundidade 406. Quando a resolução do mapa de profundidade for menor do que a resolução de uma linha de varredura de uma imagem, o módulo do mapa de profundidade 406 pode ser configurado para sobreamostrar (*up-sample*) o mapa de profundidade para casar a resolução do mapa de profundidade com a resolução da linha de varredura. Quando o mapa de profundidade não for sincronizado temporalmente com uma linha de varredura de uma imagem, o módulo de mapa de profundidade 406 pode ser configurado para sincronizar o mapa de profundidade e a linha de varredura. O módulo de mapa de profundidade 406 pode prover subsequentemente o mapa de profundidade processado para o controlador de início de linha 310, de modo que o controlador de início de linha 310 possa controlar a operação do pipeline de percurso de dados 304. Mais genericamente, uma função aritmética pode ser aplicada condicionalmente ou com base na comparação da profundidade em um local de pixel para um ou mais limiares usando um comparador, ou alternadamente, diretamente usando um bit de controle binário associado a cada pixel que pode ser aplicado no lugar da saída de comparador usando um multiplexador de bypass.

[00164] Tradicionalmente, aceleradores de hardware para operações de processamento de imagem incluem um conjunto

fixo de funções de processamento de sinais de imagem (ISP) cabeados (*hard-wired*), dispostas em uma ordem predeterminada. A Figura 5 ilustra um acelerador de hardware tradicional. Um acelerador de hardware tradicional 500 iria receber uma imagem de um dispositivo de memória 504 e processa a imagem recebida usando as funções ISP 502A-502H na ordem em que são predeterminada no momento do projeto. No exemplo mostrado na Figura 5, o acelerador de hardware 500 usa 8 funções ISP na ordem ilustrada para processar a imagem recebida. Esta abordagem é bastante inflexível e pode limitar as áreas de aplicação no qual o acelerador de hardware 500 pode ser usado. A tecnologia de sensor de imagem está se movendo rapidamente e é difícil imaginar usar um único pipeline de ISP fixo para todos os sensores atuais e futuros. Ademais, quando uma função de ISP opera um filtro em múltiplas linhas de varredura de uma imagem, a função de ISP deve armazenar, em um buffer, linhas de varredura de entrada até que o número suficiente de linhas de varredura estiver presente. Esses buffers são implementados geralmente usando dispositivos RAM dimensionados de acordo com a resolução da imagem, e o tamanho do buffer é predeterminado no momento do projeto do acelerador de hardware 500. Portanto, o buffer para o ISP pode forçar efetivamente um limite rígido na resolução de imagem que pode ser tratada pelo acelerador de hardware 500. Além disso, uma vez que o buffer é privativo para a função de ISP, o buffer não pode ser usado em outros cenários (por

exemplo por software) e pode consumir uma grande quantidade de área de pastilha.

[00165] Em algumas modalidades, o acelerador de hardware 104 aborda a inflexibilidade dos aceleradores de hardware tradicionais por encadeamento de funções de ISP comuns, genéricas. Frequentemente, a diferença entre aceleradores de hardware reside não tanto na funcionalidade das funções de ISP implementadas pelos aceleradores de hardware, mas a ordem (e em alguns casos, no número de vezes) em que as funções de ISP são invocadas. Portanto, o acelerador de hardware 104 pode ser configurado para realizar uma função desejada pelo encadeamento de um ou mais módulos de função comum, genérica, que são implementados de forma eficiente.

[00166] Por exemplo, uma operação de convolução pode ser representada como uma multiplicação e uma soma. Da mesma forma, uma operação de filtragem de resposta de impulso finita (FIR) também pode ser representada como uma multiplicação e uma soma, embora a ordem na qual as operações de filtragem FIR realiza a multiplicação e a soma pode ser diferente da operação de convolução. Apesar da diferença entre a operação de convolução e a operação de filtragem FIR, a operação de multiplicação e a operação de soma são as funções comuns para a operação de convolução e a operação de filtragem FIR. Portanto, o acelerador de hardware 104 pode ser projetado para realizar a operação de convolução e a operação de filtragem de resposta ao impulso finito usando o mesmo módulo de multiplicação e o mesmo módulo de soma.

[00167] Em algumas modalidades, a ordem na qual as funções comuns, genéricas são invocadas pode ser determinada usando software. Por exemplo, software pode programar o acelerador de hardware para invocar o módulo de multiplicação e o módulo da soma para realizar a operação de convolução ou a operação de filtragem FIR ao encadear o módulo de multiplicação e o módulo de soma em uma ordem diferente.

[00168] A Figura 6 ilustra um acelerador de hardware baseado em funções genéricas de acordo com algumas modalidades. O acelerador de hardware 102 pode incluir uma pluralidade módulos de função ISP genéricos 602A-602H, um módulo de receptor de dados 604 para receber uma ou mais linhas de varredura de uma imagem para processamento e um módulo de saída de dados 606 para emitir uma ou mais linhas de varredura que foram processadas por um ou mais módulos de função ISP genéricos 602A-602H. Em algumas modalidades, o um ou mais módulos de função de ISP genéricos 602A-602H podem incluir um registrador de configuração e um registrador de controle. Os valores para esses registros podem ser controlados usando software. Em algumas modalidades, a pluralidade de módulos de função ISP genéricos 602A-602H pode ser uma parte do pipeline de percurso de dados 304.

[00169] Em algumas modalidades, um ou mais dos módulos de função ISP genéricos 602A-602H podem incluir um filtro de hardware autocontido que também inclui capacidade de acesso direto à memória (DMA). O um ou mais módulos de função ISP genéricos 602A-602H podem usar as capacidades de DMA para carregar e/ou armazenar

dados a partir de/ou para uma fatia de memória no tecido de memória 106. As capacidades de DMA podem ser controladas usando software.

[00170] Em algumas modalidades, o módulo de receptor de dados 604 pode incluir um módulo de DMA para recuperar uma ou mais linhas de varredura de uma imagem. Em outras modalidades, o módulo de receptor de dados 604 pode incluir um módulo de interface de sensor tal como um módulo de MIPI. Em algumas modalidades, o módulo de saída de dados 606 pode incluir um módulo de DMA para armazenar uma ou mais linhas de varredura processadas de uma imagem. Em outras modalidades, o módulo de saída de dados 606 pode incluir um dispositivo de exibição.

[00171] Em algumas modalidades, o acelerador de hardware 102 pode ser acoplado a um tecido de memória 106 que inclui uma tabela de ISP. A tabela de ISP pode incluir um ou mais buffers 608. Cada buffer pode incluir um ponteiro para um dos módulos de função de ISP genéricos 602A-602H. Desde que o tecido de memória 106 possa incluir uma memória comum (ou uniforme) com múltiplas portas, diversos dispositivos podem acessar o um ou mais buffers 608 na tabela ISP para identificar módulos de função de ISP genéricos disponíveis.

[00172] Em algumas modalidades, as funções de ISP de software 612A-612C, executando em um processador 610, pode ser projetado para executar um ou mais módulos de função de ISP genéricos 602A-602H no acelerador de hardware 102. Por exemplo, uma função ISP de software 612A pode determinar (1) uma lista de módulos de função de ISP genéricos 602A-602H a ser executado para realizar

uma função desejada e (2) uma ordem na qual a lista de módulos de função de ISP genéricos 602A-602H deve ser executada. Então, a função de ISP de software 612A pode usar um ou mais buffers 608 correspondes aos módulos de função de ISP genéricos 602A-602H para encadear os módulos de função de ISP genéricos, desse modo, realizando a função desejada. Em essência, a funcionalidade do acelerador de hardware pode ser determinada pelo software em sua consulta por buffers 608 na tabela de ISP.

[00173] Em algumas modalidades, uma interface de entrada de módulos de função de ISP podem ser acoplados diretamente a uma interface de saída de outros módulos de função de ISP por meio de um buffer de primeiro a entrar, primeiro a sair (FIFO) mapeado em memória pequena. A Figura 7 ilustra um acelerador de hardware que inclui um buffer FIFO para comunicação entre os módulos de função ISP de acordo com algumas modalidades. Os módulos de função de ISP 602 podem ser acoplados a uma interface de barramento de memória 702, que por sua vez é acoplado a um buffer FIFO 704 e um tecido de memória 106.

[00174] Quando um primeiro módulo de função ISP 602A conclui sua operação em uma linha de varredura de uma imagem, o primeiro módulo de função de ISP 602A pode armazenar a linha de varredura processada em um buffer FIFO 704. Conforme o primeiro módulo de função ISP 602A continua a processar linhas de varredura adicionais, o primeiro módulo de função de ISP 602A pode continuar a armazenar as linhas de varredura processadas no buffer

FIFO 704, até que o buffer FIFO 704 esteja cheio. Quando o buffer FIFO 704 estiver cheio, o primeiro módulo de função ISP 602A pode ser parado até que o buffer FIFO 704 já não esteja cheio. Entretanto, um segundo módulo de função ISP 602B pode recuperar linhas de varredura processadas a partir do buffer FIFO 704 para processamento adicional, até que o buffer FIFO 704 esteja vazio. Com efeito, o primeiro módulo de função ISP 602A pode ser considerado o produtor de dados; o segundo módulo de função ISP 602B pode ser considerado o consumidor de dados; e o buffer FIFO 704 pode ser considerado um árbitro. Uma vez que o segundo módulo de função ISP 602B pode recuperar linhas de varredura processadas a partir do buffer FIFO 704, que tem uma latência mais baixa em comparação com uma fatia de memória no tecido de memória 106, o buffer FIFO 704 pode reduzir a latência de uma cadeia de módulos de função ISP 602.

[00175] Em algumas modalidades, o dispositivo de computação 100 pode incluir uma pluralidade de ilhas de energia. Cada ilha de energia pode ser associada a um domínio de energia dedicada. Portanto, a voltagem de alimentação de cada ilha de energia pode ser controlada independentemente. Por exemplo, o dispositivo de computação 100 pode determinar quais ilhas de energia são necessárias para realizar uma determinada operação e ligar a voltagem de alimentação apenas daquelas ilhas de energia que sejam necessárias. Desta forma, o dispositivo de computação 100 pode reduzir a fuga de consumo de energia.

[00176] Em algumas modalidades, quando o dispositivo de computação 100 determina que uma ilha de energia está atualmente em um modo de baixo consumo de energia (por exemplo, nenhuma voltagem de alimentação é provida), e que a ilha de energia é necessária para uma determinada operação, o dispositivo de computação 100 pode invocar uma sequência de energização para a ilha de energia e fornecer uma voltagem de alimentação para a ilha de energia.

[00177] Em algumas modalidades, cada um dos processadores vetoriais 102 pode ser associado a uma única ilha de energia. Em algumas modalidades, o acelerador de hardware 104 pode ser associado a uma única ilha de energia. Em algumas modalidades, o tecido de memória 106 pode ser associado a uma única ilha de energia. Em algumas modalidades, o dispositivo periférico 108 pode ser associado a uma única ilha de energia.

[00178] Em algumas modalidades, o dispositivo de computação 100 pode invocar uma sequência de energização, ao fornecer um sinal de habilitação para a ilha de energia. O sinal de habilitação pode subsequentemente fechar comutadores localizados entre a voltagem de alimentação e a ilha de energia, provendo assim a voltagem de alimentação para a ilha de energia. Esta operação é algumas vezes referida como realizar o *gating* de fornecimento de alimentação.

[00179] A Figura 8 ilustra o *gating* de fornecimento de alimentação a uma ilha de energia de acordo com algumas modalidades. A Figura 8 mostra uma ilha de energia 802,

que pode incluir blocos de circuito para o processamento de dados de entrada, um ou mais comutadores 804A-804B para fornecer uma voltagem de alimentação ou um sinal de terra para a ilha de energia 802 e um registrador de entrada 806 para reter os dados de entrada até que a ilha de energia 802 esteja pronta para processar os dados de entrada. Em algumas modalidades, o registrador de entrada 806 é acionado para fornecer os dados de entrada para a ilha de energia 802 quando o registrador de entrada 806 recebe um sinal válido recebido a partir de um gerador de sinal válido 808, indicando que a ilha de energia 802 está pronta para processar os dados de entrada.

[00180] Em algumas modalidades, o dispositivo de computação 100 é configurado para gerar um sinal válido, indicando que a voltagem de alimentação da ilha de energia atingiu uma voltagem de operação apropriada. O sinal válido pode indicar um momento no qual circuitos na ilha de energia podem ser usados para realizar as operações desejadas. O sinal válido pode ser gerado pelo gerador de sinal válido 808.

[00181] O gerador de sinal válido 808 poderia gerar o sinal válido usando um temporizador. Por exemplo, o gerador de sinal válido 808 pode determinar um momento no qual o sinal de habilitação é aplicado para a ilha de energia, e esperar uma quantidade pré-determinada de tempo usando um temporizador e então gerar o sinal válido. No entanto, a determinação da quantidade de tempo pré-determinada no momento do projeto é difícil, devido ao fato de que a quantidade de tempo que demora a

subir até a voltagem de alimentação de uma ilha de energia seria sujeito às variações de processo, voltagem e temperatura (PVT). Para lidar com as variações de PVT, o período de tempo predeterminado é muitas vezes definido de modo conservador (por exemplo, para ser suficientemente grande) para acomodar os cantos de PVT de pior caso, que pode acrescentar latência desnecessária à sequência de energização.

[00182] Para abordar estas questões, em algumas modalidades, o gerador de sinal válido 808 é configurado para gerar o sinal válido adaptativamente. Mais particularmente, a ilha de energia pode ser configurada para gerar o sinal válido ao retardar adaptativamente o sinal de habilitação fornecido para a ilha de energia.

[00183] A Figura 9 ilustra um gerador de sinal válido de acordo com algumas modalidades. O gerador de sinal válido 808 pode incluir uma pluralidade de comutadores de energia configurados para fornecer uma voltagem de alimentação para as células de lógica acopladas à pluralidade de comutadores de energia. Em algumas modalidades, os comutadores de energia podem ser uma parte de cada célula de lógica. Por exemplo, os comutadores de energia podem incluir um ou mais dispositivos de canal P em série com a alimentação positiva e/ou um ou mais dispositivos de canal N, em série com a alimentação negativa (terra). Estes comutadores de energia podem ser distribuídos ao longo de todo o bloco lógico, que compreende a ilha de energia. Na Figura 9, para simplificar, os comutadores de energia de canais N e P são mostrados como um único

bloco comutador de energia associado a cada célula lógica.

[00184] Em algumas modalidades, o gerador de sinal válido 808 pode aplicar o sinal de habilitação à cadeia de comutadores de energia de encadeamento em série e esperar até que o sinal de habilitação chegue ao fim da cadeia de comutadores de energia de encadeamento em série. Uma vez que o sinal de habilitação chega ao fim da cadeia de comutadores de energia de encadeamento em série, então assegura-se que todas as células de lógica na ilha de energia estão corretamente ligadas. Portanto, o gerador de sinal válido 808 pode usar o sinal de habilitação, retardado pela cadeia de encadeamento em série de comutadores de energia, como o sinal válido. Esse mecanismo de autocalibração pode capturar adaptativamente quaisquer variações de processo, voltagem e temperatura (PVT) do dispositivo de computação específico. Desta forma, o dispositivo de computação não precisa esperar desnecessariamente um longo período de tempo para uma ilha de energia se energizar; o dispositivo de computação pode esperar apenas a quantidade de tempo necessária para energizar a ilha de energia adequadamente.

[00185] Em algumas modalidades, uma ilha de energia pode ser sempre ligada. Em outras palavras, uma ilha de energia pode ser projetada para não entrar em um modo de baixo consumo de energia, em que é fornecida voltagem de alimentação. Uma ilha de energia pode ser referida como uma ilha de energia sempre ligada.

[00186] Em algumas modalidades, uma ilha de energia sempre ligada pode ser usada para monitorar sinais externos. Por exemplo, a ilha de energia sempre ligada pode ser usada para monitorar os sinais dos pinos de entrada e saída de uso geral (GPIO - General-Purpose-Input-Output), interfaces externas e/ou blocos funcionais internos, tal como um temporizador de baixa frequência ou reset de inicialização. Desta forma, o dispositivo de computação 100 pode analisar sinais externos, determinar se uma ou mais ilhas de energia precisam ser energizadas para responder aos sinais externos, e energizar de modo adaptativo apenas as ilhas de energia que sejam necessárias para responder aos sinais externos.

[00187] A Figura 10 ilustra um mecanismo de monitoramento de sinal de evento de acordo com algumas modalidades. A Figura 10 mostra uma ilha de energia 802 sempre ligada e um módulo de gerenciamento de energia 110. A ilha de energia 802 sempre ligada pode incluir um domínio de energia para o dispositivo periférico 108. Visto que a ilha de energia 802 sempre ligada não entra em um modo de baixo consumo de energia, o dispositivo periférico 108 na ilha de energia 802 sempre ligada pode monitorar os sinais que são assíncronos com um clock do dispositivo de computação 100. Quando o dispositivo periférico 108 detectar um sinal de evento ao qual o dispositivo de computação 100 deve responder, o dispositivo periférico 108 pode alertar o módulo de gerenciamento de energia 110. Por sua vez, o módulo de gerenciamento de energia 110 pode determinar qual das

ilhas de energia no dispositivo de computação 100 deve ser ligada. Subsequentemente, o módulo de gerenciamento de energia 110 pode fazer com que uma ou mais das ilhas de energia sejam ligadas.

[00188] Em algumas modalidades, o dispositivo periférico 108 pode incluir uma interface definida em software, cuja funcionalidade pode ser definida usando software. Mais particularmente, os dispositivos periféricos 108 podem incluir um módulo de emulação de protocolo de interface (IPE) que é capaz de emular a funcionalidade de protocolos de interface padronizados, tais como SPI, I2C, I2S ou qualquer outro protocolo adequado. A interface definida em software é benéfica porque o dispositivo periférico 108 pode manter somente uma única interface definida em software que pode ser programada para acomodar uma pluralidade de protocolos de interface, em vez de manter uma pluralidade de interfaces cada uma dedicada a um protocolo de interface específico. Uma vez que a única interface definida em software pode consumir uma área de pastilha muito menor em comparação a uma pluralidade de interfaces dedicadas, a única interface definida em software pode reduzir drasticamente os custos associados a interfaces.

[00189] A Figura 11 mostra uma interface definida em software de acordo com algumas modalidades. A Figura 11 mostra uma interface definida em software que inclui uma interface de entrada/saída (I/O) genérica 1104, um módulo IPE 1106 e um barramento interno 1108 para um dispositivo de computação 100. A interface de entrada/saída genérica 1104 pode incluir uma interface

para comunicação com um dispositivo externo, tal como um sensor ou um módulo de câmera.

[00190] A funcionalidade de uma interface de I/O 1104 pode ser configurada usando um módulo IPE 1106. Por exemplo, quando um módulo IPE 1106 determina que a interface de I/O 1104 deve operar como uma interface I2C, então o módulo IPE 1106 pode programar a interface de I/O 1104 para usar o protocolo de interface I2C para comunicação com o dispositivo externo. Em algumas modalidades, o módulo IPE 1106 pode ser programado usando software. O módulo IPE 1106 pode ser programado para que o módulo IPE 1106 possa configurar a interface de I/O 1104 para implementar protocolos de interface padronizada, tal como SPI, I2C, I2S ou quaisquer outros padrões adequados.

[00191] A Figura 12 mostra uma implementação detalhada de uma interface definida em software de acordo com algumas modalidades. A interface definida em software 1102 pode incluir uma interface de entrada/saída de uso geral (GPIO) 1202 e seus registradores 1204. Um processador host pode controlar a operação da GPIO 1202 ao configurar bits nos registradores GPIO 1204. A GPIO 1202 pode controlar alguns dos pinos na interface de I/O 1104 para se comunicar com dispositivos externos, tal como um acelerômetro, um sensor de luz ambiente ou um sensor de áudio.

[00192] A interface definida em software 1102 também pode incluir um módulo IPE 1106 e seus registradores 1206. Um processador host pode controlar a operação do módulo IPE 1106 pela configuração de bits nos

registradores de IPE 1206. O módulo de IPE 1106 pode ser configurado para determinar (1) um protocolo de interface a ser implementado pela interface definida em software 1102 e (2) pinos de interface de I/O a serem usados para implementar o protocolo de interface. Uma vez que o módulo de IPE 1106 determina os pinos de interface de I/O a serem usados para implementar o protocolo de interface, o módulo de IPE 1106 pode enviar um sinal de controle para um multiplexador 1208 multiplexar os pinos de interface de I/O selecionados para o módulo IPE 1106. O módulo IPE 1106 pode fazer com que os pinos de interface de I/O emulem o protocolo de interface, fazendo com que os pinos de interface de I/O enviem sinais de controle e de dados de acordo com o protocolo de interface.

[00193] Em algumas modalidades, o temporizador 1214 e/ou o pré-dimensionador 1216 pode ser usado para converter um clock de referência de alta frequência (por exemplo, na faixa de centenas de mega hertz) para um relógio de baixa frequência (por exemplo, na faixa de centenas de quilo hertz) para fornecer um sinal de clock adequado para o IPE. Em algumas modalidades, a frequência do clock de saída do pré-dimensionador 1216 pode ser multiplicada por um valor inteiro para emular certas interfaces. Por exemplo, quando o relógio de saída do pré-dimensionador 1216 estiver operando a 500 KHz, a frequência do relógio saída do pré-dimensionador 1216 pode ser multiplicada por três para emular interfaces I2C. Desta forma, o relógio de 500 KHz pode ser usado para operar a lógica de IPE e para

amostrar os registradores de saída conectados aos pinos de I/O.

[00194] Em algumas modalidades, o módulo de IPE 1106 no dispositivo periférico 108 pode ser configurado para realizar um desvio entre os pinos de entrada e de saída da interface de I/O 1104, desse modo emulando uma entrada em um lado do dispositivo de computação 100 e uma saída no outro lado do dispositivo de computação 100, sem energizar de fato as unidades de processamento. Isso permite que um primeiro dispositivo externo, tal como um acelerômetro acoplado ao dispositivo de computação 100 através de I2C, se comunique com um segundo dispositivo externo, tal como um processador de aplicativo SoC, sem despertar as unidades de processamento do dispositivo computação 100.

[00195] A interface definida em software 1102 também pode incluir um processador de eventos 1210 e seus registradores 1212. O processador de eventos 1210 pode ser configurado para receber sinais externos e detectar quaisquer eventos em que o dispositivo de computação 100 deve responder. A funcionalidade do processador de eventos 1210 pode ser configurada usando registradores EP 1212. Em algumas modalidades, uma vez que o processador de eventos 1210 detecta um evento para responder, o processador de eventos 1210 pode determinar os processadores vetoriais 102, aceleradores de hardware 104, e/ou tecido de memória 106 necessários para responder ao evento e enviar um sinal de habilitação de energia para a ilha de energia associada aos

processadores vetoriais determinados 102, aceleradores de hardware 104, e/ou tecido de memória 106.

[00196] A Figura 13 ilustra um processador de eventos de acordo com algumas modalidades. Como discutido acima, o processador de eventos 1210 pode se comunicar com dispositivos externos e receber sinais a partir de dispositivos externos. Os sinais podem incluir amostras de áudio, valores de acelerômetro, valores de sensor de luz ambiente ou quaisquer outras entradas que possam ser fornecidas através de uma interface de comunicação, tal como uma GPIO. O processador de eventos 1210 pode ser configurado para comparar os sinais recebidos a uma configuração específica para reconhecer um evento ou uma sequência de eventos. Uma vez que o processador de eventos 1210 reconhece um evento ou uma sequência de eventos, o processador de eventos 1210 pode fazer com que um ou mais componentes no dispositivo de computação 100 desperte de um modo de baixo consumo de energia e inicie a operação.

[00197] Em algumas modalidades, o processador de eventos 1210 pode incluir um ou mais filtros de eventos 1302A-1302N. Um filtro de eventos 1302 é configurado para receber um sinal de entrada a partir de uma interface 1104 e determinar se ocorreu um evento específico. Se o evento específico ocorreu, o filtro de eventos 1302 pode enviar um sinal de controle e/ou uma habilitação de ilha de energia a uma de uma pluralidade de ilhas de energia no dispositivo de computação 100.

[00198] A Figura 14 mostra uma implementação de um filtro de eventos de acordo com algumas modalidades. O

filtro de eventos 1302 pode incluir um registrador 1402, um comparador 1404 e um operador booleano 1406. O filtro de eventos 1302 pode ser controlado através de registradores de controle do processador de eventos 1212 e um temporizador 1214.

[00199] Os registradores de entrada 1402 podem ser configurados para receber sinais de entrada de um ou mais dispositivos externos e para fornecer os sinais de entrada recebidos em um banco de comparadores 1404. Os comparadores 1404 podem ser configurados para suportar uma ampla faixa de representações de sinal de entrada, incluindo representações booleanas, inteiras, de ponto fixo e de ponto flutuante.

[00200] Subsequentemente, as saídas dos comparadores 1404 podem ser logicamente combinadas com base no valor de temporizador a partir do temporizador de EP 1214 a fim de determinar se um determinado evento ou uma sequência de eventos ocorreu. Em alguns casos, o evento específico ou a sequência de eventos é considerada como tendo acontecido quando uma relação específica entre as saídas de comparador persistem por um período de tempo predeterminado. Uma vez que o filtro de eventos 1302 determina que um evento específico ou uma sequência de eventos aconteceu, o filtro de eventos 1302 pode emitir sinais de controle para controlar outros componentes no dispositivo de computação 100, tais como um processador vetorial 102 ou um acelerador de hardware 104 ou dispositivos externos acoplados ao dispositivo periférico 108.

[00201] O processador de eventos 1210 pode ser configurado para detectar um evento no qual um usuário inicia a usar um dispositivo eletrônico. O processador de eventos 1210 pode ligar subsequentemente componentes no dispositivo de computação 100 para responder ao evento de inicialização. Por exemplo, o processador de eventos 1210 pode ser configurado para detectar uma mudança na luz ambiente, que pode indicar que o dispositivo eletrônico foi removido de um bolso. Quando a luz ambiente continua em um alto nível por mais de alguns milissegundos, o processador de eventos 1210 pode verificar a entrada de áudio para determinar se há uma mudança nos sinais de áudio de entrada. Quando o processador de eventos 1210 detectar uma mudança nos sinais de áudio de entrada, o processador de eventos 1210 pode permitir que um processador de sinais digitais no dispositivo de computação 100 detecte um comando falado. Desta forma, o processador de eventos 1210 permite que os componentes no dispositivo de computação 100 permaneçam em um modo de baixo consumo de energia e realizem operações somente quando um evento ou uma sequência de eventos ocorreu. Portanto, o processador de eventos 1210 pode reduzir significativamente a potência de standby média do dispositivo de computação 100.

[00202] A Figura 15 mostra um modo de bypass de um dispositivo periférico de acordo com algumas modalidades. Na Figura 15, o dispositivo de computação 100 pode estar em um modo de operação de baixo consumo de energia no qual uma ou mais ilhas de energia estão em um modo de baixo consumo de energia (por exemplo, não há

voltagem de alimentação aplicada às uma ou mais ilhas de energia). Neste caso, o módulo de IPE 1106 pode ser configurado para realizar um desvio entre os pinos de entrada e de saída da interface de I/O 1104, tal como uma pista de MIPI de entrada 1502 e uma pista de MIPI de saída 1504. Neste exemplo, a pista de MIPI de entrada 1502 é acoplada a um módulo de câmera e a pista MIPI de saída 1504 é acoplada a um processador de aplicativo. Portanto, o módulo de câmera pode ser acoplado ao processador de aplicativo sem realmente despertar as uma ou mais ilhas de energia que estão em um modo de baixo consumo de energia.

[00203] Em algumas modalidades, dispositivos periféricos 108 para protocolos de interface diferentes podem compartilhar pinos físicos (ou *pads*) do dispositivo de computação 100. Por exemplo, os dispositivos periféricos 108 podem incluir uma primeira interface para um primeiro protocolo de comunicação e uma segunda interface para um segundo protocolo de comunicação. A primeira interface e a segunda interface podem ser configuradas para multiplexação no tempo de pinos de I/O físicos de modo que o número de pinos de I/O dedicados aos dispositivos periféricos 108 possa ser reduzido. Em alguns casos, os dispositivos periféricos 108 podem incluir uma tabela que inclui um mapeamento entre sinais nas primeira e segunda interfaces e pinos físicos.

[00204] Nos aplicativos onde o dispositivo de computação 100 está conectado a uma faixa de dispositivos de MIPI, tais como câmeras e telas, ou a um

processador de aplicativo ou outros dispositivos em o dispositivo de computação 100 “aparece” como uma câmera, a configuração do dispositivo de computação 100, em termos do número de blocos de interface MIPI e pinos associados, pode não ser conhecida no momento do projeto. Por esta razão, é vantajoso conectar um conjunto de pinos de I/O de MIPI a uma pluralidade de blocos de controle de protocolo de I/O MIPI programáveis de modo que o número de entradas e saídas MIPI requeridas para suportar um caso de uso de MIPI específico possa ser configurado em tempo de execução através de software.

[00205] A Figura 16 mostra uma interface de MIPI programável de acordo com algumas modalidades. A interface de MIPI programável 1600 pode incluir um bloco de protocolo de controle de acesso à mídia (MAC) MIPI 1602, um transmissor de MIPI 1604, um receptor de MIPI 1606, um multiplexador 1608 que é configurado para sinais de canal de qualquer um dos transmissores de MIPI 1604 ou o receptor de MIPI 1606, um comutador de polaridade MIPI 1610 que é configurado para mudar a polaridade dos *pads* de I/O de MIPI diferenciais 1612, e um multiplexador de bypass 1614 e um buffer de bypass 1616 para realizar um desvio entre pinos de entrada e pinos de saída da interface de I/O 1104, conforme ilustrado com relação à Figura 15.

[00206] Em algumas modalidades, o bloco de protocolo MAC de MIPI 1602 é projetado para controlar a operação do transmissor de MIPI 1604 e/ou do receptor de MIPI 1606, de modo que a operação do transmissor de MIPI 1604

e/ou do receptor de MIPI 1606 conforme-se com o protocolo MIPI.

[00207] Em algumas modalidades, a interface de MIPI programável 1600 pode permitir que somente um transmissor de MIPI 1604 ou o receptor de MIPI 1606 se comunique através do *pad* de I/O de MIPI 1612 em um momento específico. Por exemplo, a interface de MIPI programável 1600 pode acoplar somente um dentre o transmissor de MIPI 1604 ou o receptor de MIPI 1606 com o *pad* de I/O de MIPI 1612 através de um multiplexador 1608. Desta forma, para um dispositivo externo, o *pad* de I/O de MIPI 1612 pode ser considerado uma interface MIPI bidirecional.

[00208] Em algumas modalidades, a interface MIPI programável 1600 pode usar o comutador de polaridade de MIPI 1610 para inverter a polaridade dos *pads* de I/O de MIPI diferenciais para que a polaridade dos *pads* de I/O de MIPI diferenciais possa ser invertida no tempo de execução de modo a atingir melhor casamento de impedância ou para corrigir erros no projeto de PCB externo sem retrabalho. A Figura 17 ilustra um aplicativo de um mecanismo de inversão de polaridade para uma interface de entrada/saída de acordo com algumas modalidades. Enquanto a Figura 17 ilustra a aplicação do mecanismo de inversão de polaridade para *pads* de I/O de MIPI, o mecanismo de inversão de polaridade pode ser usado em uma variedade de outras interfaces que usam par diferencial de linhas de sinal.

[00209] Em algumas modalidades, conforme descrito de modo geral acima com relação à Figura 15, a interface de

MIPI programável 1600 pode fornecer um modo de bypass de MIPI de baixo consumo de energia ao fornecer multiplexadores de MIPI 1614 e buffers 1616 que permitem *pads* de I/O de MIPI 1612 sejam conectados às saídas sem requerer que as unidades de processamento do dispositivo computação 100 sejam energizadas. Esta característica é desejável nos modos onde múltiplos sensores de câmera estão conectados ao dispositivo de computação 100 para realizar tarefas de visão por computador enquanto em outros casos de uso do dispositivo de computação 100 não é requerido e o processador de aplicativo realiza captura de imagem fixa ou de vídeo usando o mesmo conjunto de sensores. Com o fornecimento de multiplexadores de MIPI internos 1614, tais casos de uso podem ser suportados através dos multiplexadores de bypass internos 1614, ao invés de usar componentes externos e simplifica grandemente o custo e a complexidade da PCB em que os chips são combinados.

[00210] Em algumas modalidades, o tecido de memória 106 pode incluir memória cache que é projetada para explorar os dados localmente, em termos de ambas as localidades espacial e temporal. Quando um dispositivo de computação 100 não for acoplado a um dispositivo de memória externa, o tecido de memória 106 pode permitir que processadores vetoriais 102 e aceleradores de hardware 104 usem a memória cache como um dispositivo de memória geral. Em algumas modalidades, a memória cache pode ser particionada em seções de modo que cada seção é utilizada exclusivamente por um dos processadores vetoriais ou um dos aceleradores de hardware.

[00211] Em algumas modalidades, o tecido de memória 106 é configurado para manter informações de estado do dispositivo de computação 100 quando o dispositivo de computação 100 estiver em uma modo de economia de energia. Desta forma, quando o dispositivo de computação 100 for ligado novamente, o dispositivo de computação 100 pode redistribuir as informações de estado de dispositivos apropriados para que o retardo associado ao procedimento de “despertar” (“wake-up”) possa ser reduzido.

[00212] Em alguns casos, as informações de estado são mantidas na memória cache. Em tais casos, a memória cache que armazena as informações de estado pode ser energizada mesmo quando o dispositivo de computação 100 entrar em uma modo de economia de energia. As informações de estado podem incluir binários de aplicativo(s) carregadas na inicialização ou durante o tempo de execução. As informações de estado também podem incluir informações de configuração, tais como configurações de registrador, modo de operação, configuração de pipeline e configurações do ambiente de tempo de execução carregado no momento da inicialização e modificado durante o tempo de execução que teria, de outro modo, ser armazenado em memória não-volátil externa e recuperada no evento de uma sequência de desenergização e energização. As informações de estado também podem incluir dados tais como dados de imagem e valores de outros sensores. As informações de estado também podem incluir o estado de protocolos de comunicação entre o dispositivo de computação 100 e

outros componentes do sistema que caso contrário precisariam ser armazenados e recuperados da memória não-volátil externa no evento de uma sequência de desenergização e energização.

[00213] Em algumas modalidades, o tecido de memória 106 pode incluir um controlador baseado em hardware de exclusão mútua (mutex) 206. A Figura 18 ilustra um tecido de memória tendo um controlador mutex baseado em hardware de acordo com algumas modalidades. A Figura 18 mostra uma pluralidade de unidades de processamento 1802A-1802P, um tecido de memória 106 e um controlador mutex 206. Uma unidade de processamento 1802 pode incluir um processador vetorial 102 ou um acelerador de hardware 104. O controlador mutex 206 pode incluir um ou mais elementos mutex de endereçáveis de modo independente que são configurados para coordenar multitarefa de unidades de processamento 1802 que compartilham de um elemento de dados. Mais particularmente, um elemento de mutex pode ser configurado para bloquear um elemento de dados compartilhados, armazenado no tecido de memória 106 ou outras partes do dispositivo de computação 100, para uma primeira unidade de processamento 1802A de modo que outras unidades de processamento 1802P também usam o elemento de dados compartilhado, possam esperar até que a primeira unidade de processamento 1802A libere o elemento de dados compartilhados. Devido ao fato de que o controlador mutex 206 reside dentro do tecido de memória 106, o tempo de liberação ou bloqueio de um

recurso compartilhado é reduzido quando comparado ao uso de um barramento compartilhado ou outros meios.

[00214] Tradicionalmente, quando um controlador mutex recebe uma requisição para um acesso exclusivo a um recurso compartilhado, o controlador mutex responde imediatamente à requisição, indicando se a unidade de processamento pode obter um acesso exclusivo ao recurso compartilhado. Portanto, se a unidade de processamento requerente não provê um acesso exclusivo, a unidade de processamento requerente tem que requisitar o controlador mutex continuamente até que a unidade de processamento requerente receba o acesso exclusivo do controlador mutex. Isto pode aumentar o tráfego no barramento entre o controlador mutex tradicional e as unidades de processamento.

[00215] Para atender este problema, em algumas modalidades, quando uma unidade de processamento 1802A envia uma requisição de acesso exclusivo, requisitando que um acesso exclusivo a um recurso compartilhado, o controlador mutex 206 pode monitorar o status da requisição por conta própria. Uma vez que o controlador mutex 206 determina que para a unidade de processamento 1802A é concedido o acesso exclusivo, o controlador mutex 206 pode enviar uma mensagem de confirmação para a unidade de processamento 1802A, indicando que a unidade de processamento 1802A tem o acesso exclusivo ao recurso compartilhado. Desta forma, não é requerido que a unidade de processamento 1802A envie a requisição de acesso exclusivo múltiplas vezes até que a unidade de processamento 1802A receba o acesso exclusivo; a unidade

de processamento 1802A pode enviar a requisição de acesso exclusivo apenas uma vez e esperar para receber o acesso exclusivo a partir do controlador mutex 206. Esse mecanismo de mensagem pode reduzir a carga de comunicação sobre o tecido de memória 106.

[00216] Em algumas modalidades, o tecido de memória 106 pode incluir uma arquitetura de barramento flexível que provê comunicação entre as unidades de processamento. Frequentemente, uma interface para comunicação entre unidades de processamento inclui um buffer, tal como um primeiro a entrar, primeiro a sair (FIFO). Por exemplo, quando uma primeira unidade de processamento estiver pronta para enviar uma mensagem para uma segunda unidade de processamento, a primeira unidade de processamento pode enviar a mensagem para um buffer que é designado para a segunda unidade de processamento. Quando a segunda unidade de processamento estiver pronta para receber a mensagem, a segunda unidade de processamento pode recuperar a mensagem a partir do buffer.

[00217] No entanto, buffers em interfaces tradicionais têm uma capacidade de armazenamento limitada. Portanto, buffers em interfaces tradicionais muitas vezes estão limitados ao armazenamento de mensagens de controle e não podem acomodar uma grande quantidade de dados, tais como dados de imagem e de vídeo. Além disso, cada buffer é permanentemente designado a uma das unidades de processamento. Portanto, enquanto um primeiro buffer designado a uma primeira unidade de processamento pode estar transbordando (*overflowing*), um segundo buffer designado a uma segunda unidade de processamento pode

estar vazia. Assim, a capacidade dos buffers pode não ser totalmente utilizada no nível de sistema.

[00218] O tecido de memória 106 atende estas deficiências das interfaces tradicionais, aumentando a capacidade dos buffers e designando dinamicamente buffers para unidades de processamento com base nas demandas de comunicação em tempo real. O tecido de memória 106 provê um mecanismo flexível para criar, gerenciar e liberar buffers. Os buffers podem ser criados para a duração de um processo e podem ser liberados quando o processo for concluído. O buffer liberado pode ser disponibilizado para outros aplicativos ou unidades de processamento sob um controle de programa de software.

[00219] A Figura 19 ilustra uma atribuição dinâmica de buffers de acordo com algumas modalidades. O tecido de memória 106 pode incluir uma pluralidade de buffers 1902A-1902P, cada um dos quais pode ser designado exclusivamente a cada uma das unidades de processamento, tais como um processador vetorial ou um acelerador de hardware. Em alguns casos, múltiplos buffers 1902 podem ser designados à mesma unidade de processamento.

[00220] Em algumas modalidades, a pluralidade de buffers 1902 pode ser parte de um repositório de buffers que pode ser particionado e designado exclusivamente a uma das unidades de processamento. O repositório pode compreender uma fatia de memória a partir do tecido de memória 106. Em algumas modalidades, cada um da pluralidade de buffers 1902 pode ter a mesma capacidade. Em outras modalidades, um ou mais dos buffers 1902 pode

ter uma capacidade variável. Por exemplo, quando uma primeira unidade de processamento 1802N tenta enviar um pequeno número de mensagens de controle para uma segunda unidade de processamento 1802C, o tecido de memória 106 pode designar um buffer pequeno 1902C para a segunda unidade de processamento 1802 para que a segunda unidade de processamento 1802 possa receber o pequeno número de mensagens de controle. No entanto, quando a primeira unidade de processamento 1802N tenta enviar uma grande quantidade de dados de vídeo para a segunda unidade de processamento 1802M, o tecido de memória 106 pode designar um buffer tendo uma grande capacidade para a segunda unidade de processamento 1802M de modo que a segunda unidade de processamento 1802M possa receber a grande quantidade de vídeo.

[00221] Em algumas modalidades, uma ou mais da pluralidade de buffers 1902 podem ser associadas com aplicações específicas, tais como uma interface de comunicação, incluindo USB, MIPI ou Ethernet que podem ser previstas no dispositivo (sistema em chip) no momento do projeto.

[00222] Em algumas modalidades, o módulo de gerenciamento de energia 110 pode ser configurado para prover uma voltagem de alimentação de energia diferente para circuitos lógicos e dispositivos de memória. A Figura 20 ilustra um mecanismo de gerenciamento de energia que fornece diferentes voltagens para dispositivos de memória de circuitos lógicos de acordo com algumas modalidades. Uma única ilha de energia 2002A pode incluir uma área de circuito lógico 2004 e uma área

de memória 2006. O módulo de gerenciamento de energia 110 pode ser configurado para prover uma primeira voltagem V_1 para a área de circuito lógico 2004 e uma segunda voltagem V_2 para uma área de memória 2006. Em algumas modalidades, a primeira voltagem e a segunda voltagem podem ser providas por um regulador de energia diferente. Portanto, a primeira voltagem e a segunda voltagem podem ser controladas independentemente.

[00223] Em algumas modalidades, a área de circuito lógico 2004 e a área de memória 2006 podem entrar em um modo de baixo consumo de energia de modo independente. Por exemplo, o módulo de gerenciamento de energia 110 pode usar comutadores locais 2008, 2010 para cortar a voltagem de alimentação para a área de circuito lógico 2004 e para a área de memória 2006, respectivamente. Em algumas modalidades, o módulo de gerenciamento de energia 110 pode usar o comutador global 2012 para cortar a voltagem de alimentação para áreas de memória 2006 em uma ou mais ilhas de energia 2002A, ..., 2002N.

[00224] Em algumas modalidades, o tecido de memória 106 pode incluir um mecanismo de acesso direto à memória (DMA). O mecanismo de DMA pode manter uma lista de operações, que inclui uma lista duplamente ligada de estruturas de dados DMA. Cada estrutura de dados DMA indica uma operação específica a ser realizada pelo mecanismo de DMA. As estruturas de dados DMA são mantidas na ordem em que o mecanismo de DMA deve realizar as operações associadas às estruturas de dados DMA.

[00225] Devido ao fato da lista de operações incluir uma lista duplamente ligada de estruturas de dados DMA, é necessária uma quantidade significativa de tempo para remover uma operação de DMA para a sequência de operações representadas pela lista duplamente ligada. Em algumas modalidades, o mecanismo de DMA pode atender a esse problema, mantendo um buffer que indica se uma estrutura de dados DMA deve ser executada. Cada bit no buffer pode ser considerado um sinal de habilitação para a estrutura de dados DMA associada.

[00226] A Figura 21 ilustra um mecanismo de DMA que implementa um mecanismo de habilitação de estrutura de dados DMA baseado em buffer, de acordo com algumas modalidades. O mecanismo de DMA inclui uma lista de operações 2102 que apresenta uma pluralidade de estruturas de dados DMA 2104. A pluralidade de estruturas de dados DMA 2104 pode ser acoplada entre si como uma lista duplamente ligada. O mecanismo de DMA também inclui um buffer de habilitação 2106. O buffer de habilitação 2106 pode incluir uma pluralidade de bits. O número de bits no buffer de habilitação 2106 pode ser idêntico ao número de estruturas de dados DMA na lista de operações 2102. Cada bit no buffer de habilitação 2106 pode indicar se uma estrutura de dados DMA associada ao bit será habilitada. Por exemplo, quando um primeiro bit no buffer for um "1", então, o mecanismo DMA pode determinar que a primeira estrutura de dados DMA está habilitada e executar a primeira estrutura de dados DMA. Quando um segundo bit no buffer for um "0", o mecanismo de DMA pode determinar que a segunda estrutura

de dados DMA está habilitada e não executar a segunda estrutura de dados DMA. Desta forma, o mecanismo de DMA pode executar seletivamente um subconjunto das estruturas de dados DMA na lista de operações sem de fato remover as estruturas de dados DMA da lista de operações. Visto que o mecanismo de DMA não precisa remover as estruturas de dados DMA, o retardo associado à desabilitação de uma ou mais estruturas de dados DMA pode ser pequeno.

[00227] Em algumas modalidades, o dispositivo de computação paralelo 100 pode residir em um dispositivo eletrônico. A Figura 22 ilustra um dispositivo eletrônico que inclui o dispositivo de computação de acordo com algumas modalidades. O dispositivo eletrônico 2200 pode incluir um processador 2202, uma memória 2204, uma ou mais interfaces 2206 e o dispositivo de computação 100.

[00228] O dispositivo eletrônico 2200 pode ter uma memória 2204 tal como uma mídia legível por computador, memória flash, uma unidade de disco magnético, uma unidade óptica, uma memória programável somente de leitura (PROM), e/ou uma memória apenas de leitura (ROM). O dispositivo eletrônico 2200 pode ser configurado com um ou mais processadores 2202 que processam instruções e executam o software que pode ser armazenado na memória 2204. O processador 2202 também pode se comunicar com a memória 1404 e as interfaces 2206 para se comunicar com outros dispositivos. O processador 2202 pode ser qualquer processador aplicável tal como um sistema em chip que combina uma CPU, um

processador de aplicativo e uma memória flash, ou um processador de conjunto reduzido de instruções de computação (RISC).

[00229] A memória 2204 pode ser uma mídia legível por computador não transitória, memória flash, uma unidade de disco magnético, uma unidade óptica, uma memória somente de leitura programável (PROM), uma memória apenas de leitura (ROM), ou qualquer outra memória ou combinação de memórias. O software pode executar em um processador capaz de executar instruções de computador ou código de computador. O processador também pode ser implementado em hardware usando um circuito integrado de aplicação específica (ASIC), um arranjo lógico programável (PLA), arranjo de portas programáveis em campo (FPGA), ou qualquer outro circuito integrado.

[00230] As interfaces 2206 podem ser implementadas em hardware ou software. As interfaces 2206 podem ser usadas para receber informações de dados e de controle a partir da rede, bem como fontes locais, tais como um controle remoto para uma televisão. O dispositivo eletrônico também pode fornecer uma variedade de interfaces de usuário, tais como um teclado, uma tela sensível ao toque, uma trackball, um touch pad, ou um mouse. O dispositivo eletrônico também pode incluir alto-falantes e um dispositivo de exibição em algumas modalidades.

[00231] Em algumas modalidades, uma unidade de processamento, tal como um processador vetorial 102 e um acelerador de hardware 104, no dispositivo de computação 100, pode incluir um chip integrado capaz de executar

instruções de computador ou código de computador. O processador também pode ser implementado em hardware usando um circuito integrado de aplicação específica (ASIC), um arranjo lógico programável (PLA), arranjo de portas programáveis em campo (FPGA), ou qualquer outro circuito integrado.

[00232] Em algumas modalidades, o dispositivo de computação 100 pode ser implementado como um sistema em chip (SOC). Em outras modalidades, um ou mais blocos no dispositivo de computação paralelo podem ser implementados como um chip separado, e o dispositivo de computação paralelo pode ser empacotado em um sistema em pacote (SIP). Em algumas modalidades, o dispositivo de computação paralelo 400 pode ser usado para aplicativos de processamento de dados. Os aplicativos de processamento de dados podem incluir aplicativos de processamento de imagem e/ou aplicativos de processamento de vídeo. Os aplicativos de processamento de imagem podem incluir um processo de processamento de imagem, incluindo uma operação de filtragem de imagem; os aplicativos de processamento de vídeo podem incluir uma operação de decodificação de vídeo, uma operação de codificação de vídeo, uma operação de análise de vídeo para detecção de movimento ou objetos em vídeos. Aplicações adicionais da presente invenção incluem a aprendizagem de máquina e classificação com base na sequência de imagens, objetos ou vídeos e aplicações de realidade aumentada, incluindo aquelas onde um aplicativo de jogo extrai a geometria de múltiplas vistas de câmera incluindo câmeras habilitadas em

profundidade e extrai características das múltiplas vistas das quais a geometria de *wireframe* (por exemplo através de uma nuvem de pontos) pode ser extraída para sombreamento de vértice subsequente por uma GPU.

[00233] O dispositivo eletrônico 2200 pode incluir um dispositivo móvel, tal como um telefone celular. O dispositivo móvel pode comunicar-se com uma pluralidade de redes de acesso de rádio usando uma pluralidade de tecnologias de acesso e redes de comunicações com fio. O dispositivo móvel pode ser um telefone inteligente oferecendo capacidades avançadas, tais como processamento de texto, navegação na web, jogos, capacidades de livro digital, um sistema operacional e um teclado completo. O dispositivo móvel pode executar um sistema operacional tal como o Symbian OS, iPhone OS, RIM's Blackberry, Windows Mobile, Linux, Palm WebOS, e Android. A tela pode ser uma tela sensível ao toque que pode ser usada para entrada de dados para o dispositivo móvel e a tela pode ser usada em vez do teclado completo. O dispositivo móvel pode ter a capacidade de executar aplicativos ou de se comunicar com aplicativos que são fornecidos pelos servidores da rede de comunicações. O dispositivo móvel pode receber atualizações e outras informações a partir desses aplicativos na rede.

[00234] O dispositivo eletrônico 2200 também pode englobar muitos outros dispositivos, tais como televisores (TVs), projetores de vídeo, set-top boxes ou unidades de set-top, gravadores de vídeo digital (DVR), computadores, netbooks, laptops, computadores tablet e

qualquer outro equipamento audiovisual que possa se comunicar com uma rede. O dispositivo eletrônico também pode manter as coordenadas de posicionamento globais, informações de perfil ou outras informações de localização em sua pilha ou memória.

[00235] Será apreciado que apesar de vários arranjos diferentes terem sido descritos neste documento, as características de cada um podem ser vantajosamente combinadas juntas em uma variedade de formas para obter vantagem.

[00236] No relatório acima, o aplicativo foi descrito com referência a exemplos específicos. No entanto, será evidente que várias modificações e alterações podem ser realizadas sem partir do espírito e do escopo da invenção como reivindicado nas reivindicações anexas. Por exemplo, as conexões podem ser de qualquer tipo de conexão apropriada para transferir sinais a partir de, ou para, os respectivos nós, unidades ou dispositivos, por exemplo através de dispositivos intermédios. Nesse sentido, a menos que implícito ou afirmado em contrário, as conexões podem, por exemplo, serem conexões diretas ou conexões indiretas.

[00237] Deve ser entendido que as arquiteturas descritas aqui são meramente exemplares, e que de fato, muitas outras arquiteturas podem ser implementados as quais alcançam a mesma funcionalidade. Em um sentido abstrato, mas em um sentido definitivo ainda, qualquer arranjo dos componentes para atingir a mesma funcionalidade é efetivamente “associada” tal que a funcionalidade desejada seja alcançada. Portanto,

qualquer dois componentes combinados neste documento para atingir uma determinada funcionalidade podem ser vistos como “associados” uns aos outros tal que a funcionalidade desejada seja atingida, independentemente das arquiteturas ou componentes intermediários. Da mesma forma, quaisquer dois componentes assim associados também podem ser vistos como sendo “conectados operativamente”, ou “acoplados operativamente” uns aos outros para atingir a funcionalidade desejada.

[00238] Ademais, os versados na técnica reconhecerão que os limites entre a funcionalidade das operações descritas acima são meramente ilustrativos. A funcionalidade de múltiplas operações pode ser combinada em uma única operação, e/ou a funcionalidade de uma única operação pode ser distribuída em operações adicionais. Além disso, modalidades alternativas podem incluir múltiplos casos de uma determinada operação, e a ordem das operações pode ser alterada em várias outras modalidades.

[00239] Entretanto, outras modificações, variações e alternativas também são possíveis. O relatório e desenhos devem, por conseguinte, ser considerados em um sentido ilustrativo e não em um sentido restritivo.

[00240] Nas reivindicações, quaisquer sinais de referência dispostos entre parênteses não devem ser interpretados como limitando a reivindicação. A palavra “compreendendo” não exclui a presença de outros elementos ou etapas do que aqueles listadas em uma reivindicação. Além disso, os termos “um” ou “uma” como usado aqui, é definido como um ou mais de um. Ademais, o

uso de frases introdutórias como “pelo menos um” e “um ou mais” nas reivindicações não deve ser interpretado como significando que a introdução de outro elemento de reivindicação pelos artigos indefinidos “um” ou “uma” limita qualquer reivindicação específica que contenha tal elemento de reivindicação introduzido para invenções contendo apenas um tal elemento, mesmo quando a mesma reivindicação inclui as frases introdutórias “um ou mais” ou “pelo menos um” e os artigos indefinidos, tais como “um” ou “uma”. O mesmo vale para o uso de artigos definidos. Salvo indicação em contrário, termos como “primeiro” e “segundo” são utilizados arbitrariamente para distinguir entre os elementos que descrevem tais termos. Assim, esses termos não necessariamente pretendem indicar priorização temporal ou outra de tais elementos. O simples fato de que certas medidas são descritas em reivindicações mutuamente diferentes, não indica que uma combinação destas medidas não possa ser usada em benefício.

REIVINDICAÇÕES

1. Dispositivo de computação **(100)**
caracterizado por compreender:

uma pluralidade de processadores vetoriais (102), sendo que um primeiro processador vetorial da pluralidade de processadores vetoriais é configurado para executar uma instrução que opera uma primeira matriz de valores, o primeiro processador vetorial para receber energia a partir de uma primeira ilha de energia;

um acelerador de *hardware* (104) configurado para realizar uma operação de filtragem em uma segunda matriz de valores, o acelerador de *hardware* para receber a energia a partir de uma segunda ilha de energia;

um tecido de memória (106) incluindo uma pluralidade de fatias de memória e um sistema de interconexão (202) possuindo uma primeira interface e uma segunda interface, em que a primeira interface é configurada para acoplar a pluralidade de processadores vetoriais (102) à pluralidade de fatias de memória e em que a segunda interface é configurada para acoplar o acelerador de *hardware* (104) à pluralidade de fatias de memória;

um processador *host* configurado para fazer com que o tecido memória (106) forneça a primeira matriz de valores para o primeiro processador vetorial através da primeira interface e para fazer com que o tecido de memória forneça a segunda matriz de valores para o acelerador de *hardware* (104) através de uma

segunda interface, de modo a possibilitar que o primeiro processador vetorial processe a primeira matriz de valores de acordo com as instruções e de modo a possibilitar que o acelerador de *hardware* (104) processe a segunda matriz de valores de acordo com a operação de filtragem; e

um dispositivo periférico (108) acoplado a uma pluralidade de pinos de entrada/saída (I/O), em que o dispositivo periférico (108) configurado para fornecer um canal de comunicação entre o primeiro processador vetorial e um dispositivo externo, em que o dispositivo periférico (108) compreende um módulo de emulação que é configurado para fazer com que o dispositivo periférico (108) emule uma funcionalidade de uma pluralidade de interfaces de protocolo padrão através de um conjunto de pinos de I/O.

2. Dispositivo de computação (100), de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende uma pluralidade de ilhas de energia (2002A-2002N), cada uma compreendendo pelo menos um domínio de energia, sendo que uma primeira ilha de energia de uma pluralidade de ilhas de energia (2002A-2002N) é acoplada a uma primeira voltagem de alimentação para fornecer a primeira voltagem de alimentação para o primeiro processador vetorial de uma pluralidade de processadores vetoriais (102), e sendo que uma segunda ilha de energia de uma pluralidade de ilhas de energia (2002A-2002N) é acoplada a uma segunda voltagem de alimentação para fornecer a segunda voltagem de alimentação para o acelerador de *hardware* (104).

3. Dispositivo de computação (100), de acordo com a reivindicação 2, **caracterizado** pelo fato de que compreende um módulo de gerenciamento de energia (110) configurado para fornecer um sinal de habilitação para um comutador que acopla o primeiro da pluralidade das ilhas de energia (2002A-2002N) para a primeira voltagem de alimentação, de modo a dispor o primeiro processador vetorial da pluralidade de processadores vetoriais (102) em modo ativo.

4. Dispositivo de computação, de acordo com a reivindicação 3, **caracterizado** pelo fato de que o primeiro processador vetorial inclui uma região de circuito lógico (2004) para processar a primeira matriz de valores e uma memória local (2006) para armazenar pelo menos um subconjunto da primeira matriz de valores, em que o módulo de gerenciamento de energia (110) é configurado para fazer com que a primeira voltagem de alimentação seja fornecida para a região de circuito lógico (2004) e para fazer com que uma terceira voltagem de alimentação seja fornecida para a memória local (2006) para controlar o consumo de energia—da região de circuito lógico (2004) e da memória local (2006) independentemente.

5. Dispositivo de computação, de acordo com a reivindicação 3, **caracterizado** pelo fato de que o módulo de gerenciamento de energia (110) é configurado para desligar o comutador para desconectar a primeira ilha de energia e a primeira voltagem de alimentação, de modo a colocar o primeiro processador vetorial em um modo de baixo consumo.

6. Dispositivo de computação, de acordo com a reivindicação 3, **caracterizado** pelo fato de que o módulo de gerenciamento de energia (110) compreende um gerador de ~~um~~ sinal válido (808) configurado para gerar um sinal válido, indicando um momento no qual blocos de circuito (2004) na primeira ilha de energia estão prontos para processar dados de entrada, em que o gerador de sinal válido (808) compreende um encadeamento em série de comutadores que fornece a primeira voltagem de alimentação para os blocos de circuito (2004) na primeira ilha de energia da pluralidade de ilhas de energia (2002A-2002N).

7. Dispositivo de computação, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o dispositivo periférico (108) está associado a uma ilha de energia (2002A-2002N) que está sempre ligada.

8. Dispositivo de computação, de acordo com a reivindicação 7, **caracterizado** pelo fato de que o dispositivo periférico (108) é configurada para monitorar os sinais a partir do dispositivo externo para detectar um evento ao qual o primeiro processador vetorial deve responder, e o dispositivo periférico (108), ao detectar um evento, faz com que o módulo de gerenciamento de energia (110) coloque o primeiro processador vetorial em um modo ativo.

9. Dispositivo de computação, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o dispositivo periférico (108) é acoplado com um par diferencial de pinos de I/O, e o dispositivo periférico (108) é configurado para mudar uma polaridade do par

diferencial com base em um sinal de controle de polaridadee.

10. Dispositivo de computação, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os pinos de I/O compreendem um par diferencial de interface de processador da indústria móvel (MIPI).

11. Dispositivo de computação, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o dispositivo periférico (108) compreende um *buffer* de *bypass* (1616) que é configurada para realizar um desvio entre um pino de I/O de entrada e um pino de I/O de saída, dito desvio provê um canal de comunicação entre o pino de I/O de entrada e o pino de I/O de saída sem colocar o primeiro processador vetorial em um modo ativo.

12. Método **caracterizado** por compreender:
fornecer um tecido de memória (106) compreendendo uma pluralidade de fatias de memória e um sistema interconexão (202) possuindo uma primeira interface e uma segunda interface;

acoplar, através do uso da primeira interface, a pluralidade de fatias de memória e uma pluralidade de processadores vetoriais (102);

acoplar, através do uso da segunda interface, a pluralidade de fatias de memória e um acelerador de rádio (104);

fornecer, através de um tecido de memória, uma primeira matriz de valores para o primeiro processador vetorial através de uma primeira interface e

fornecer uma segunda matriz de valores para o acelerador de *hardware* (104) através de uma segunda interface; e

executar, no primeiro processador vetorial, uma instrução que opera na primeira matriz de valores;

realizar, através de acelerador de *hardware* (104), uma operação de filtragem em uma segunda matriz de valores;

fornecer um dispositivo periférico (108) acoplado a uma pluralidade de pinos de entrada/saída (I/O), em que o dispositivo periférico é configurado para fornecer um canal de comunicação entre pelo menos um processador vetorial e dispositivo externo, em que o dispositivo periférico é associado com uma ilha de energia (2002A-2002N) está sempre ligada; e

emular, através do dispositivo periférico (108), uma funcionalidade de uma pluralidade de interfaces de protocolo padrão através de um conjunto de pinos de entrada/saída (I/O); e

controlar, com base nas informações fornecidas por um dispositivo externo, quando a energia é fornecida para o primeiro processador vetorial através de uma primeira ilha de energia e quando a energia é fornecida para o acelerador de *hardware* através de uma segunda ilha de energia.

13. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que compreende:

fornecer uma primeira voltagem de alimentação para o primeiro processador vetorial; e

fornecer uma segunda voltagem de alimentação para o acelerador de *hardware* (104), em que o primeiro processador vetorial e o acelerador de *hardware* (104) estão associados com uma primeira ilha de energia (2002A-2002N) e uma segunda ilha de energia (2002A-2002N), respectivamente.

14. Método, de acordo com a reivindicação 13, **caracterizado** pelo fato de que o módulo de gerenciamento de energia fornece um sinal de habilitação para um comutador acoplando a primeira ilha de energia (2002A-2002N) à primeira voltagem de alimentação, de modo que o primeiro processador vetorial é colocado em um modo ativo.

15. Método, de acordo com a reivindicação 13, **caracterizado** pelo fato de que inclui adicionalmente a geração de um sinal válido indicando um momento no qual blocos de circuito (2004) na primeira ilha de energia (2002A-2002N) estão prontos para processar dados de entrada, usando um encadeamento em série de comutadores que fornece a primeira voltagem de alimentação para o bloco de circuito (2004) no primeiro processador vetorial de uma pluralidade de processadores vetoriais (102).

16. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que inclui adicionalmente o monitoramento de sinais a partir do dispositivo externo para detectar um evento ao qual o primeiro processador vetorial deve responder, fazendo com que o módulo de gerenciamento de energia (110) coloque o primeiro processador vetorial em um modo ativo.

17. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que o dispositivo periférico (108) está acoplado com um par diferencial de pinos de I/O, o método incluindo adicionalmente alterar a polaridade do par diferencial com base em um sinal de controle de polaridade.

18. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que inclui adicionalmente realizar um desvio entre um pino de I/O de entrada e um pino de I/O de saída usando um buffer de by-pass (1616), de modo a fornecer um canal de comunicação entre o pino de I/O de entrada e o pino de I/O de saída sem colocar o primeiro processador vetorial em um modo ativo.

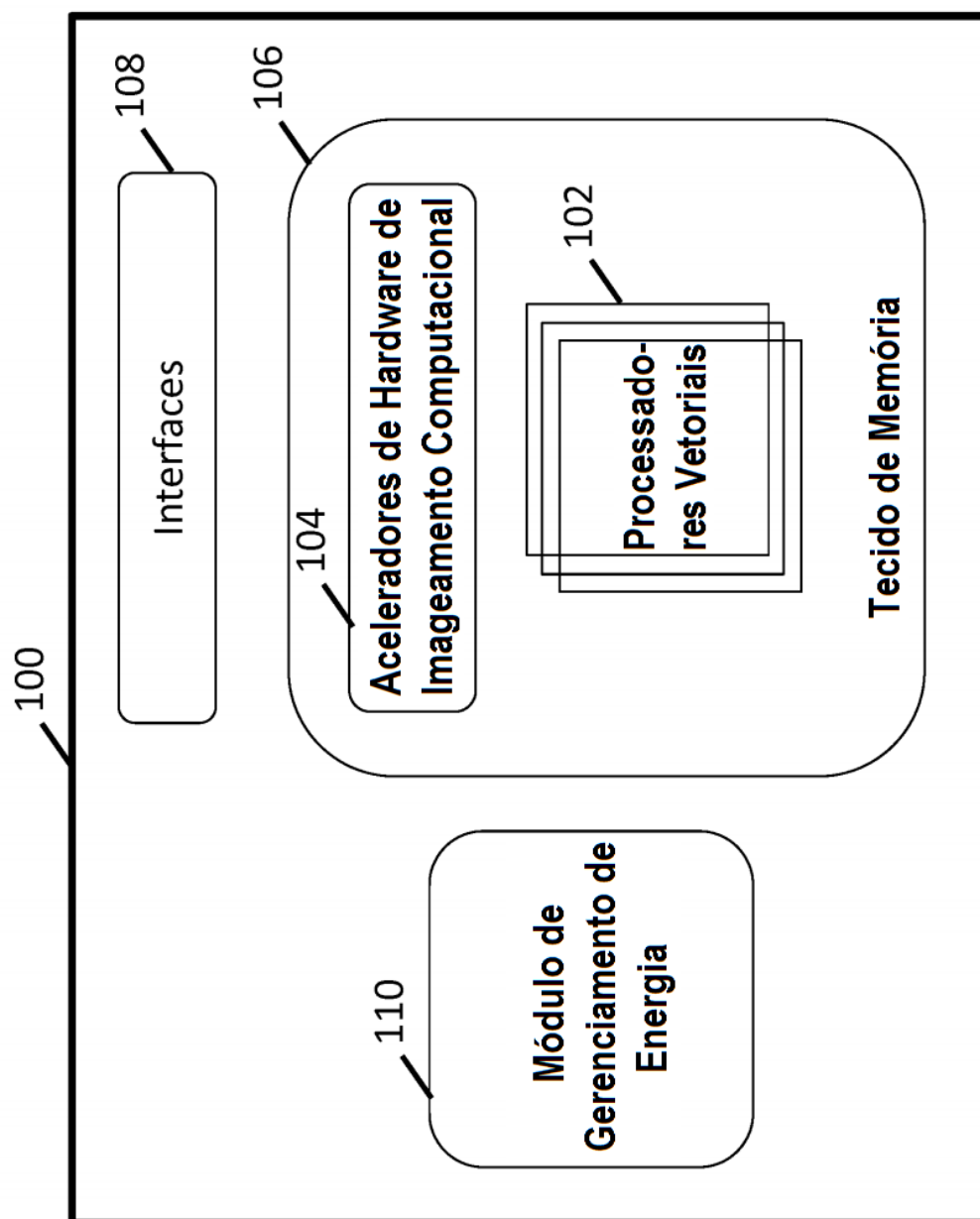


FIG. 1



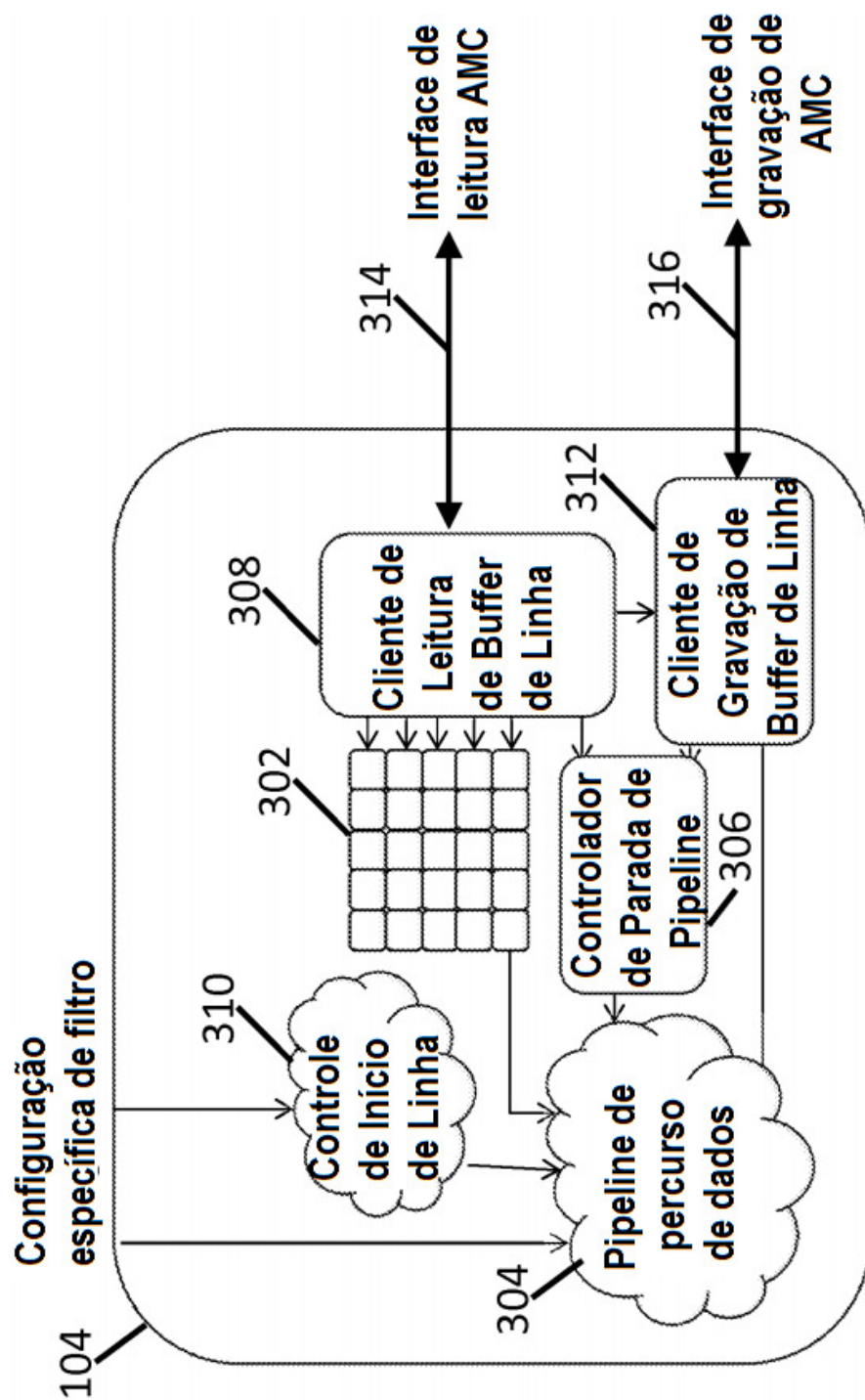


FIG. 3

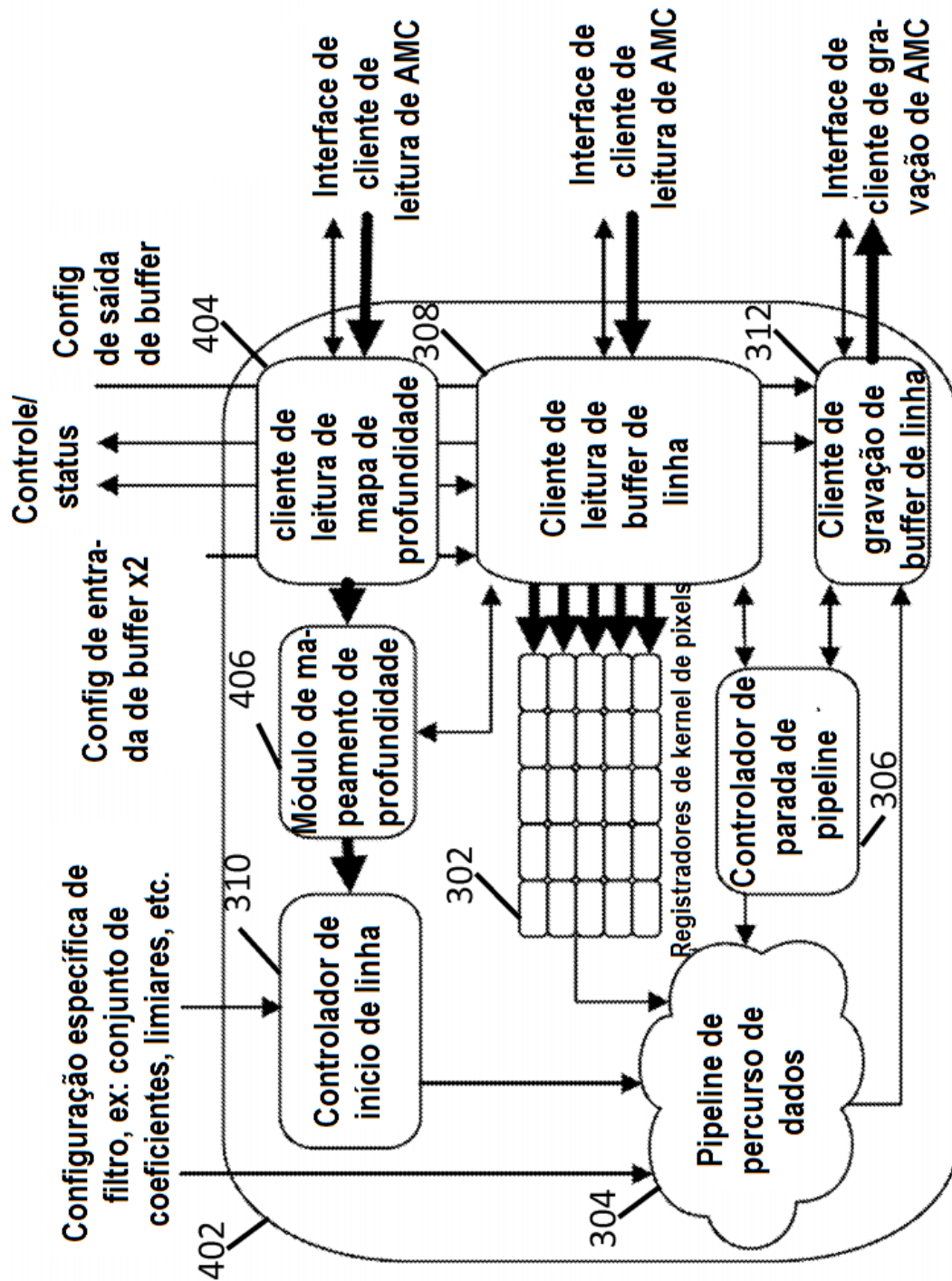


FIG. 4

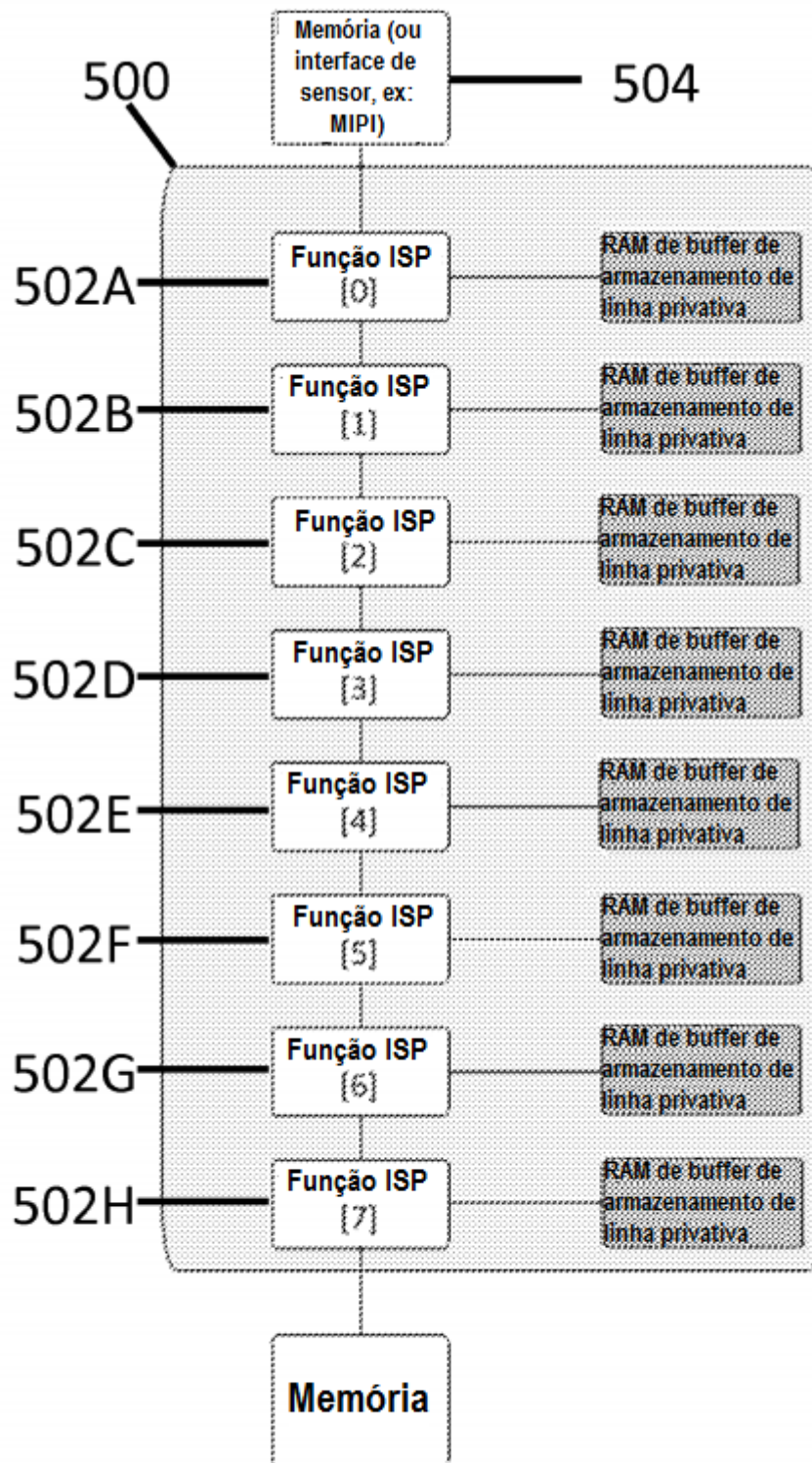


FIG. 5

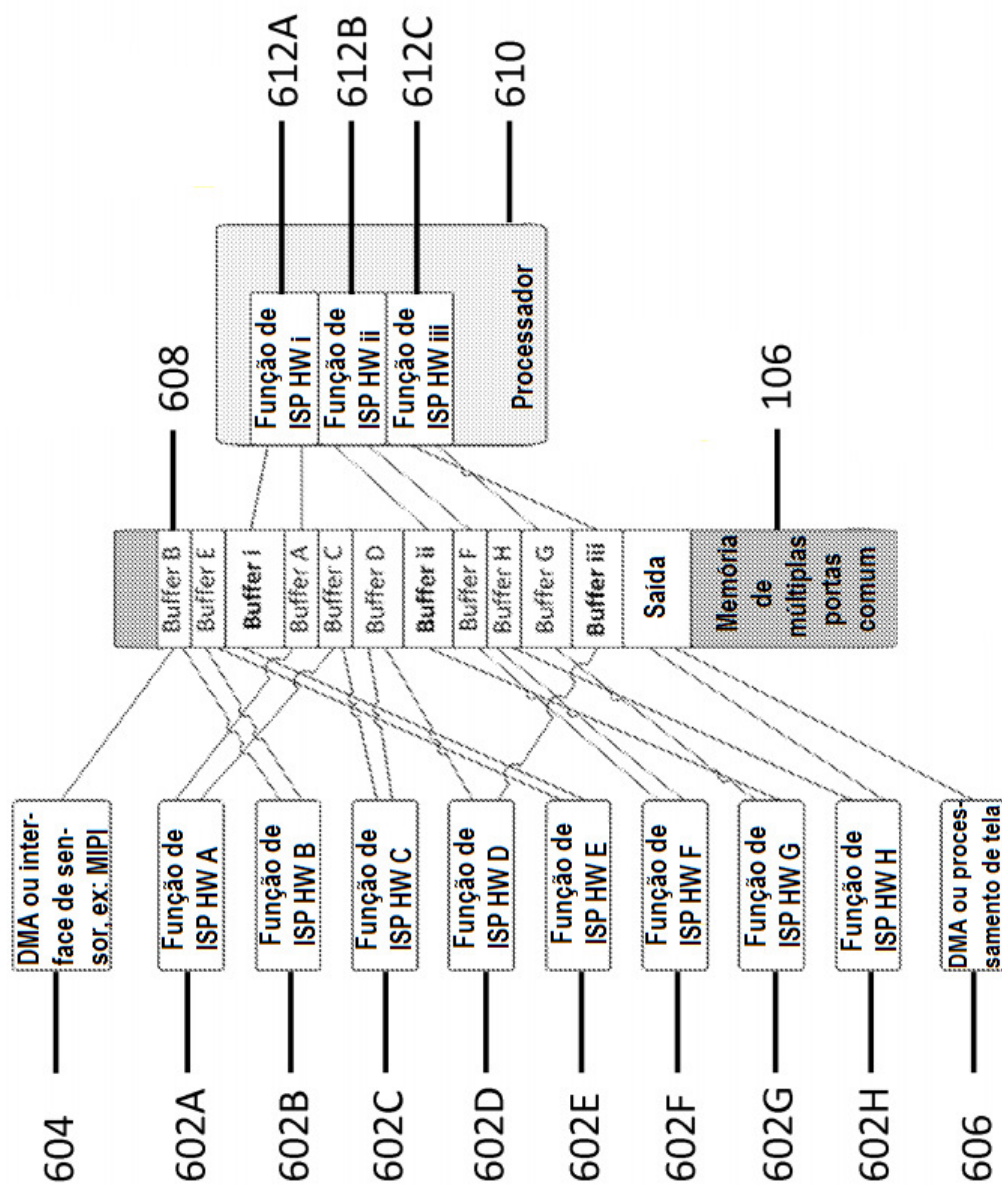


FIG. 6

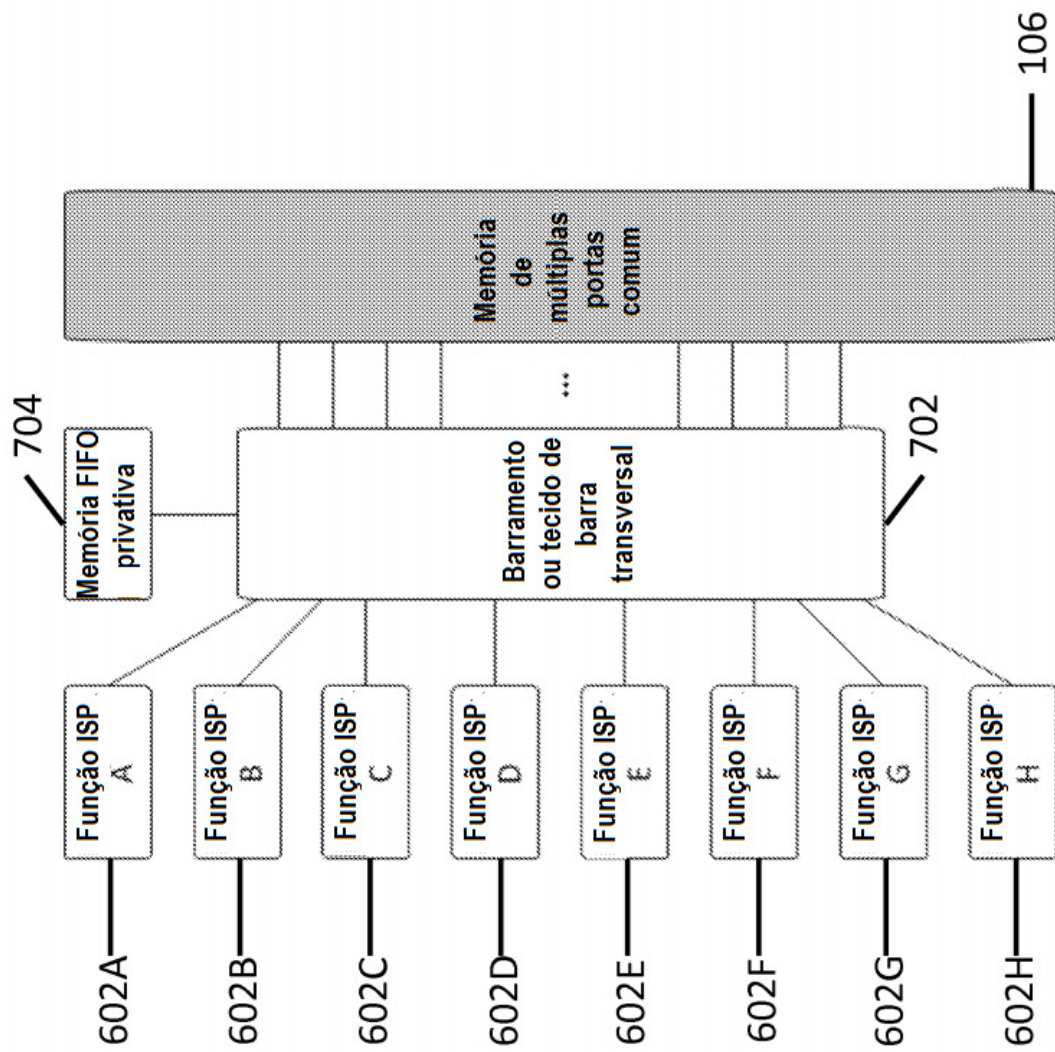


FIG. 7

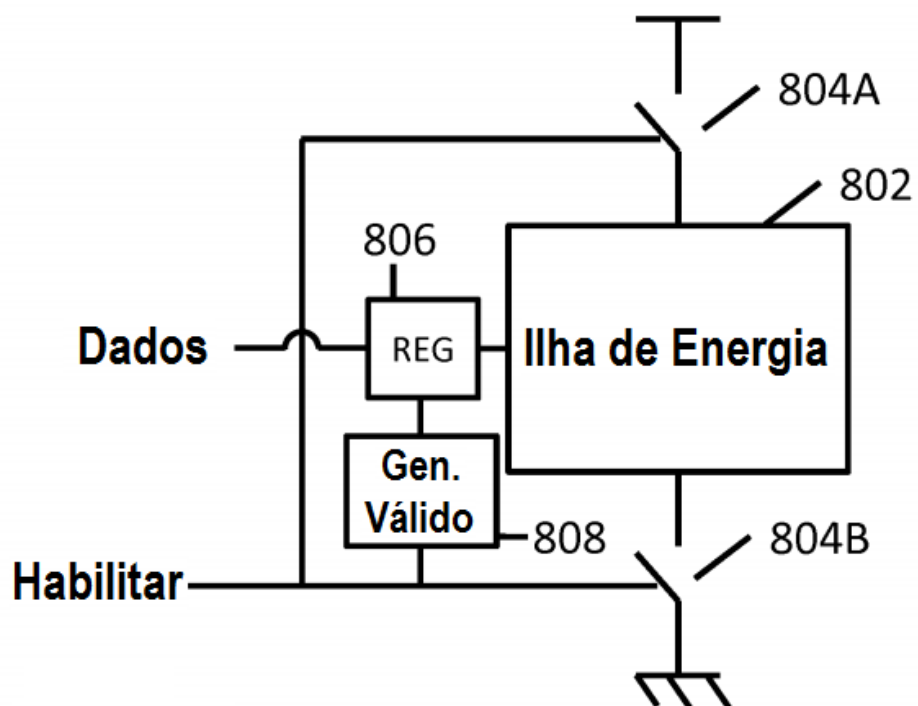


FIG. 8

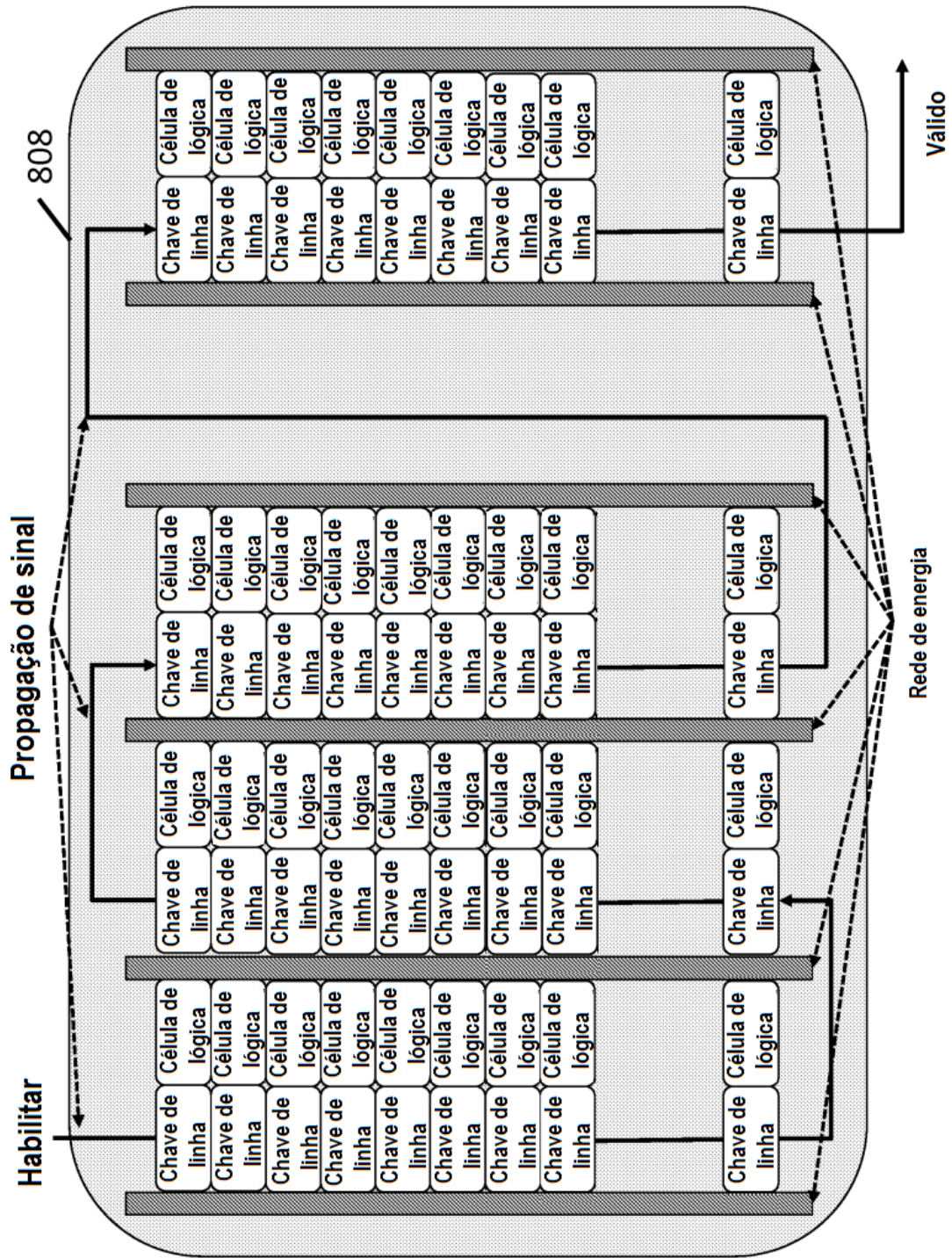


FIG. 9

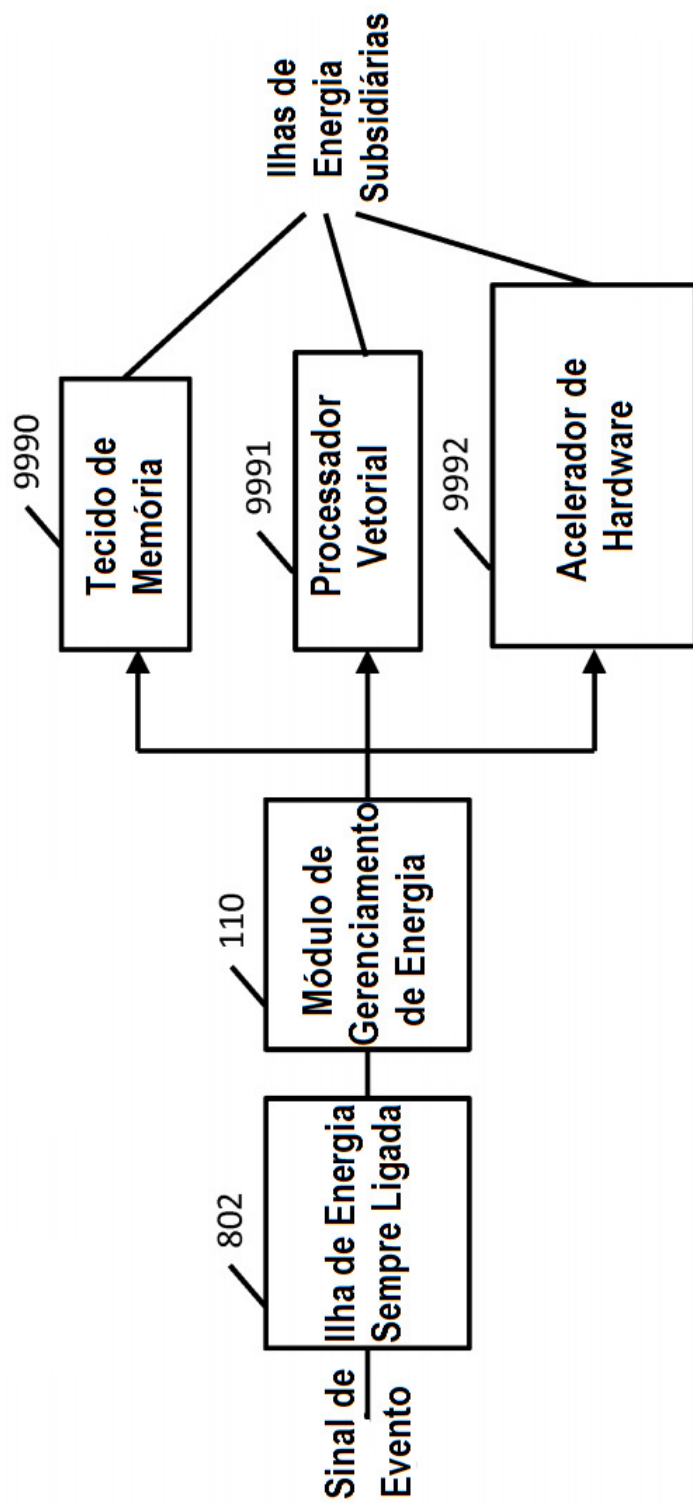


FIG. 10

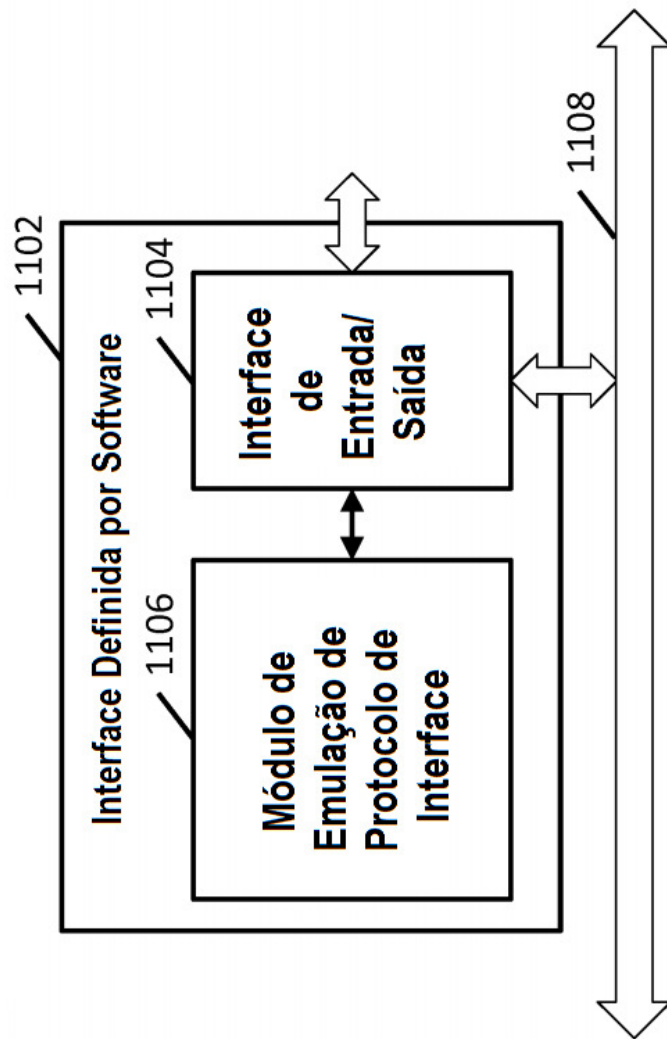


FIG. 11

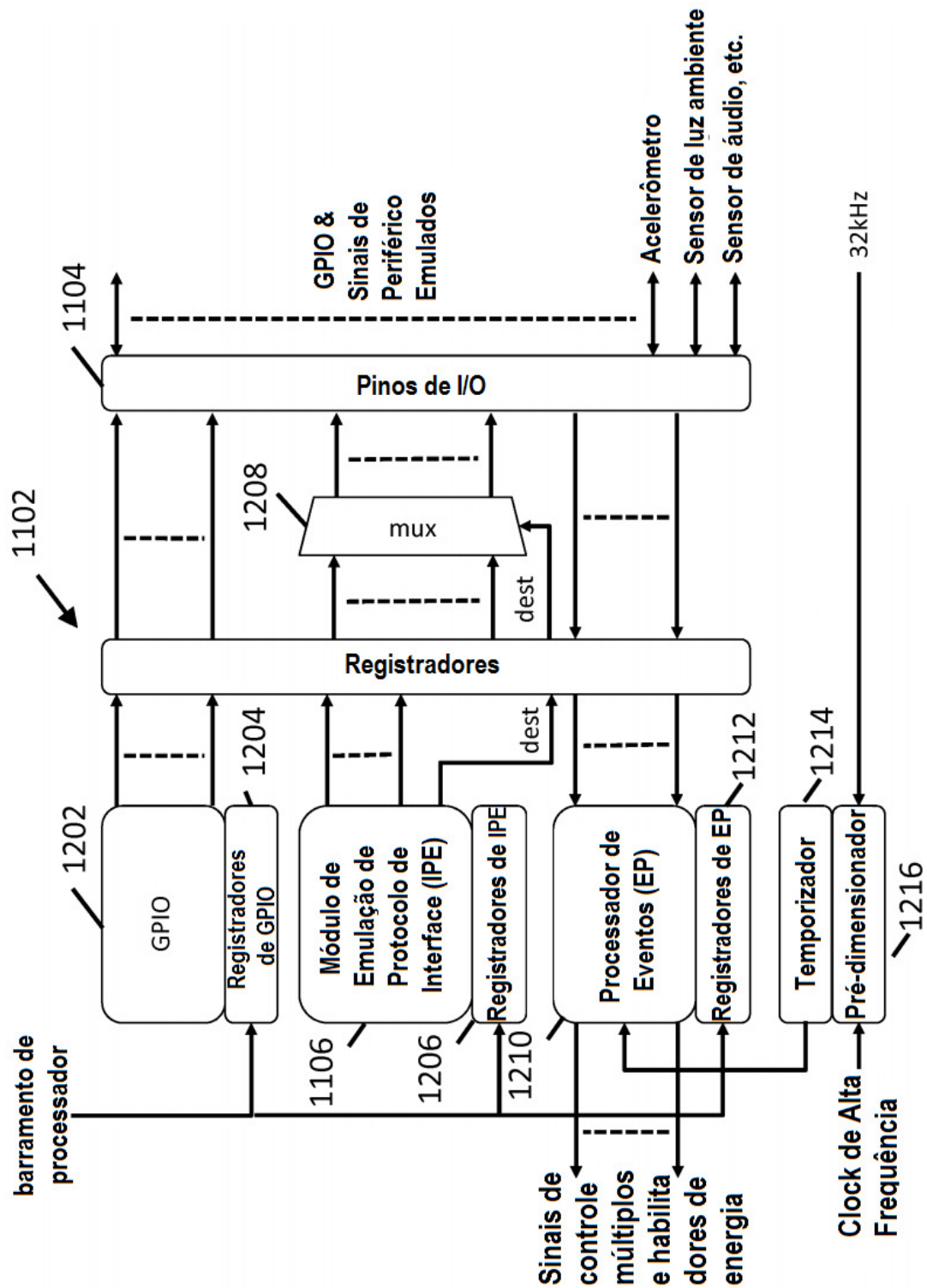


FIG. 12

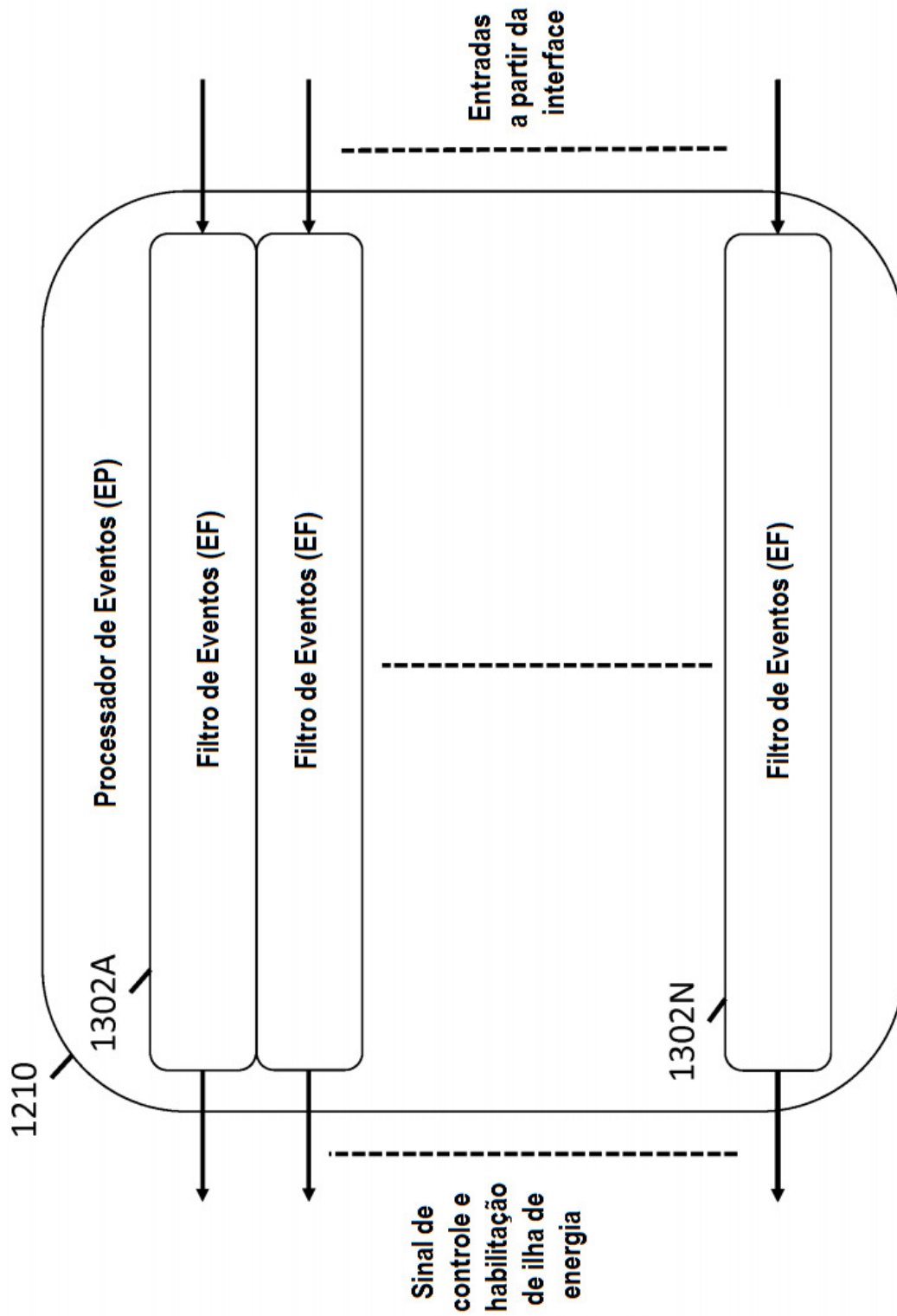


FIG. 13

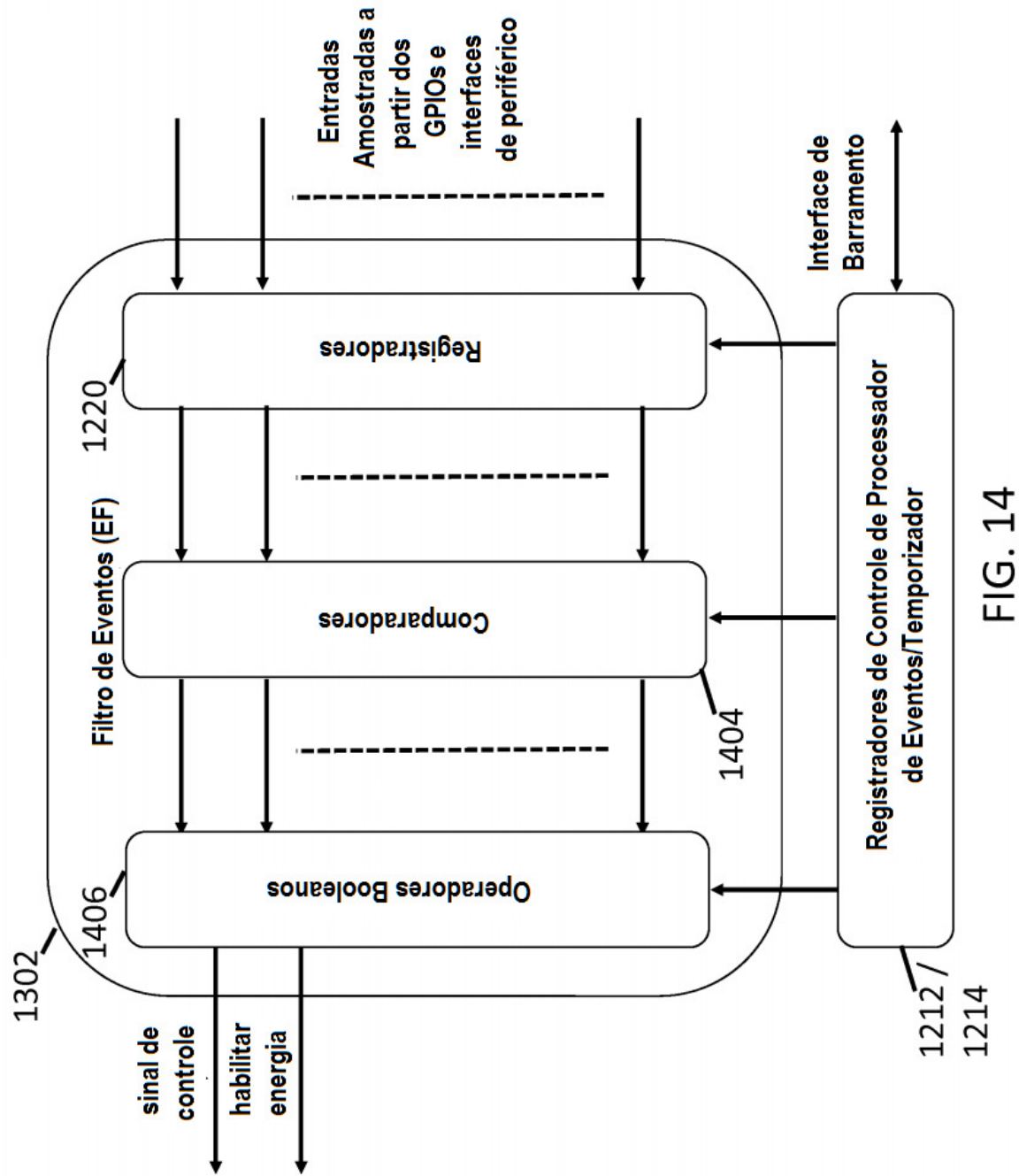


FIG. 14

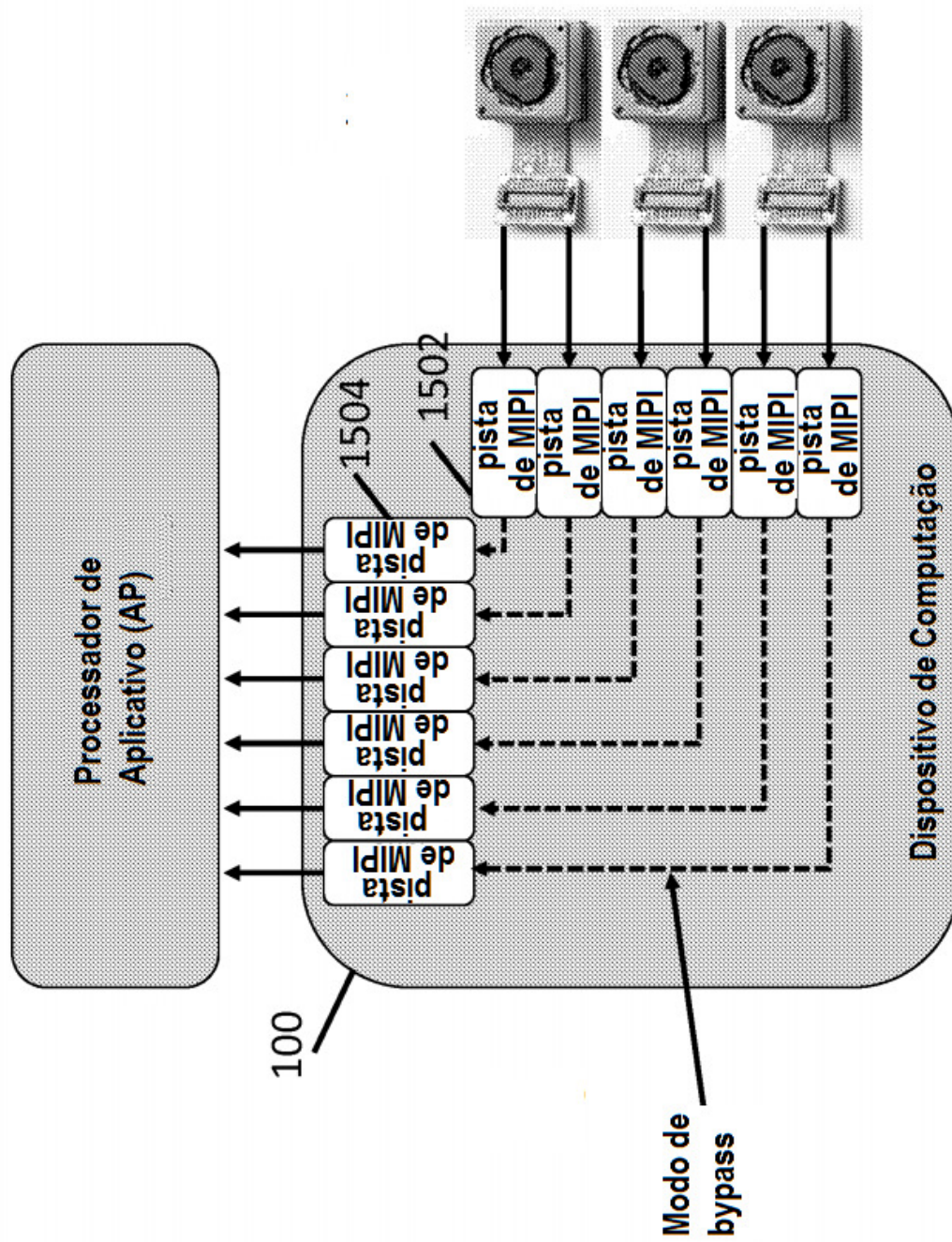


FIG. 15

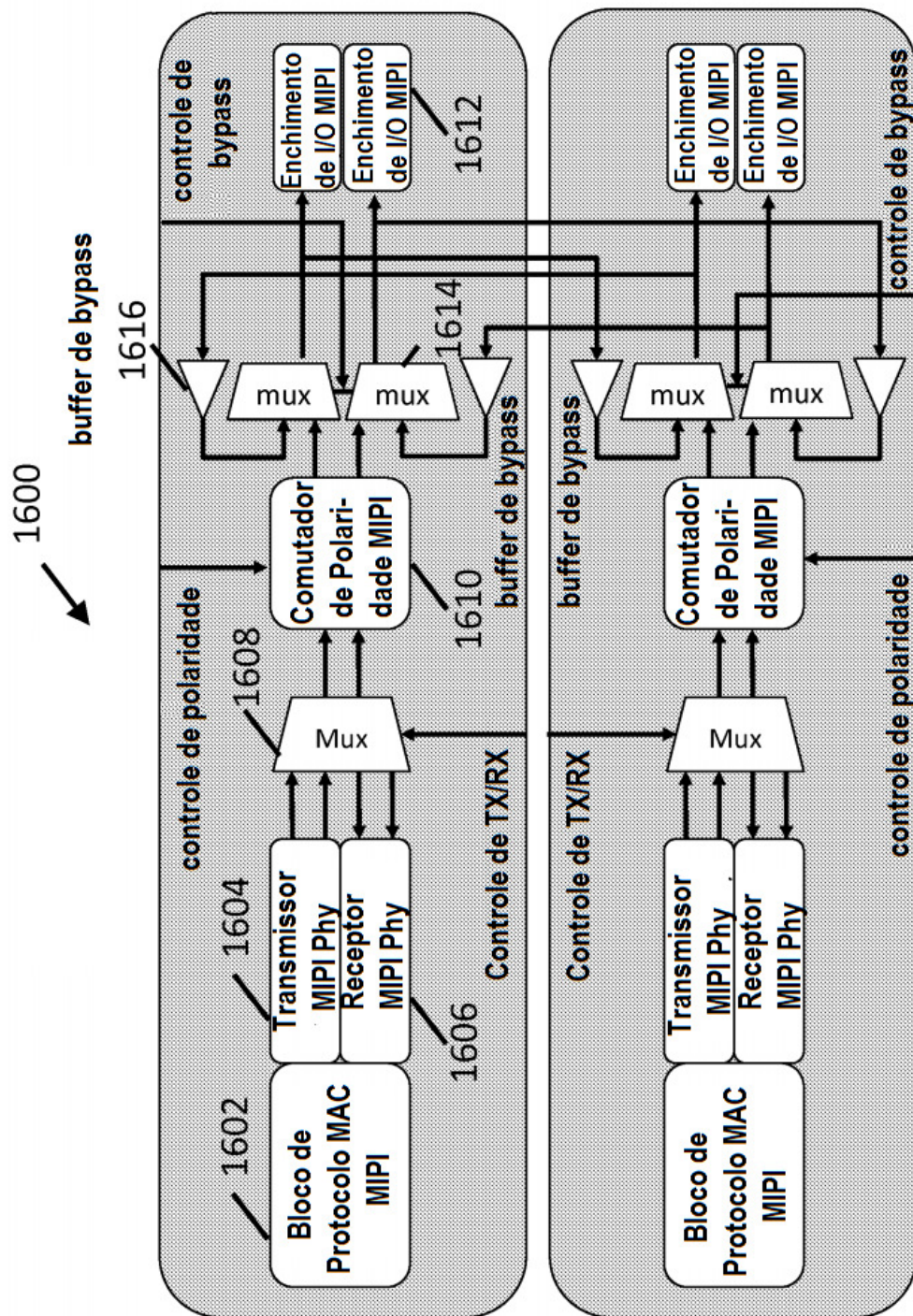


FIG. 16

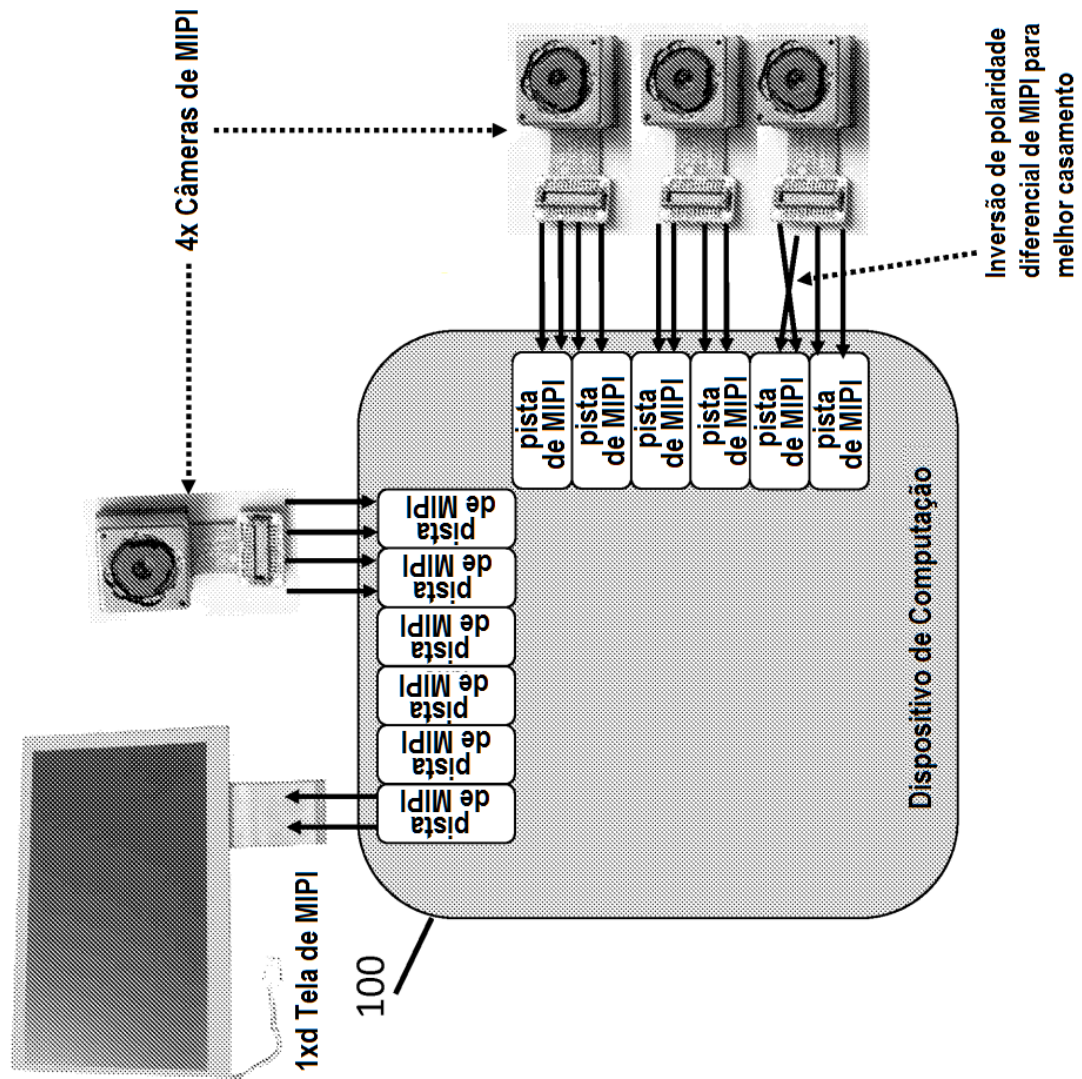


FIG. 17

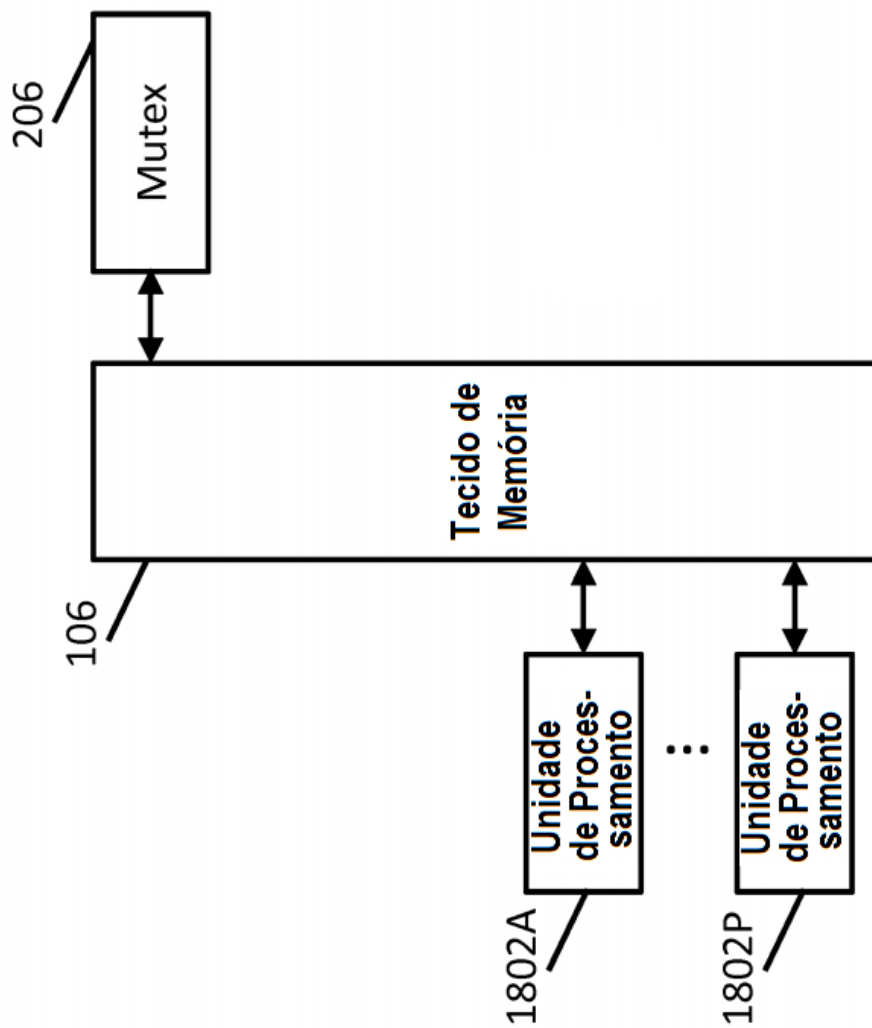


FIG. 18

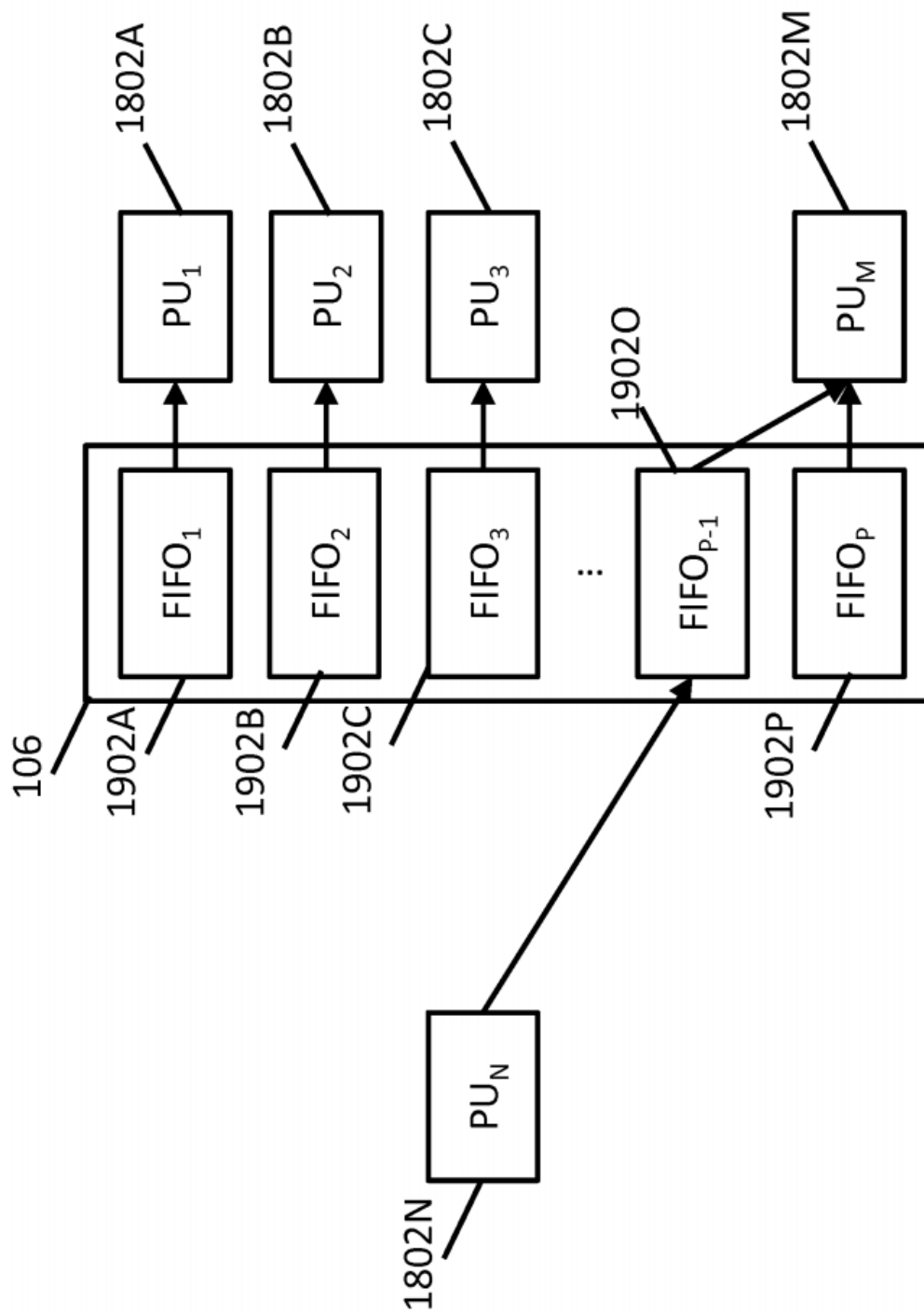


FIG. 19

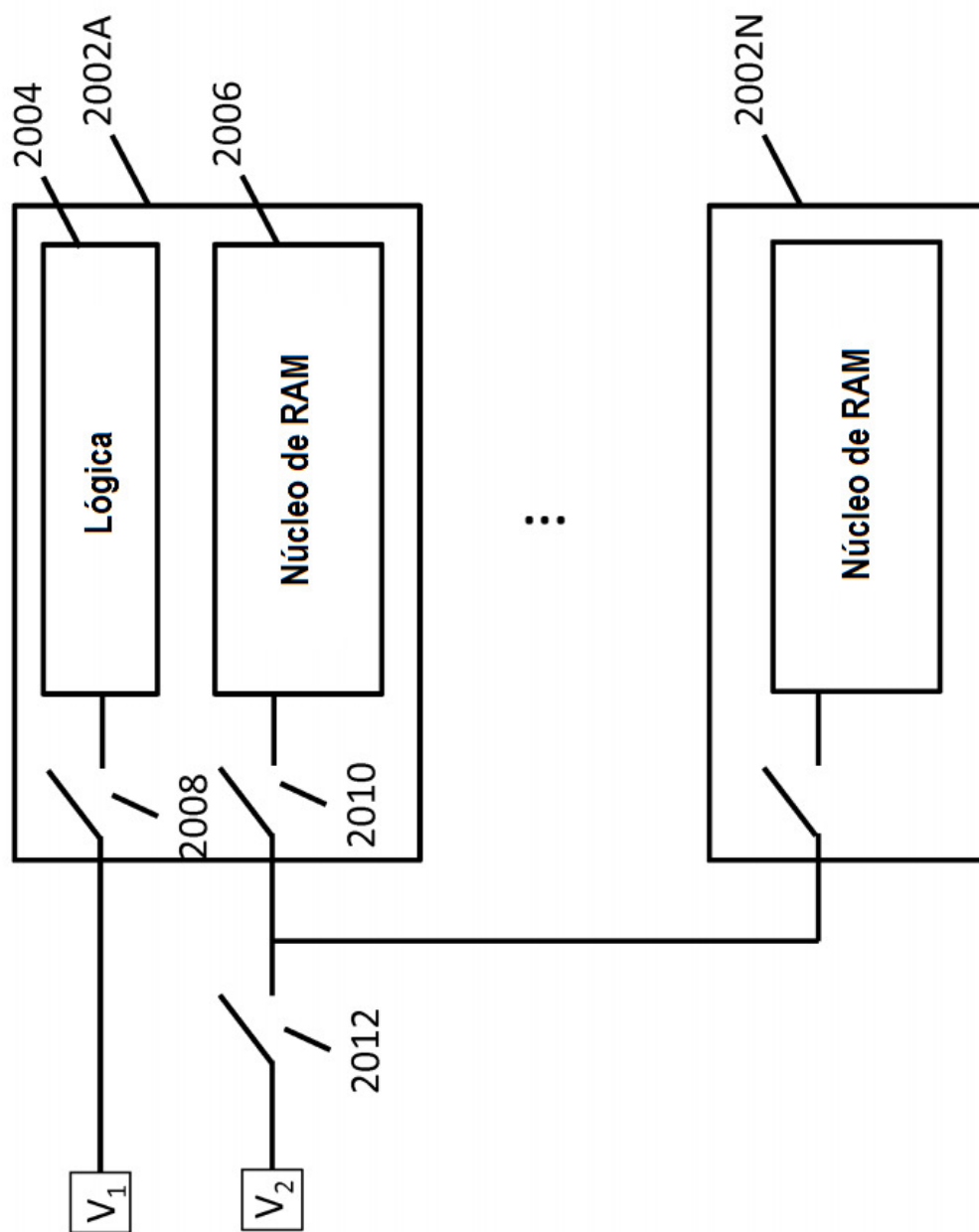


FIG. 20

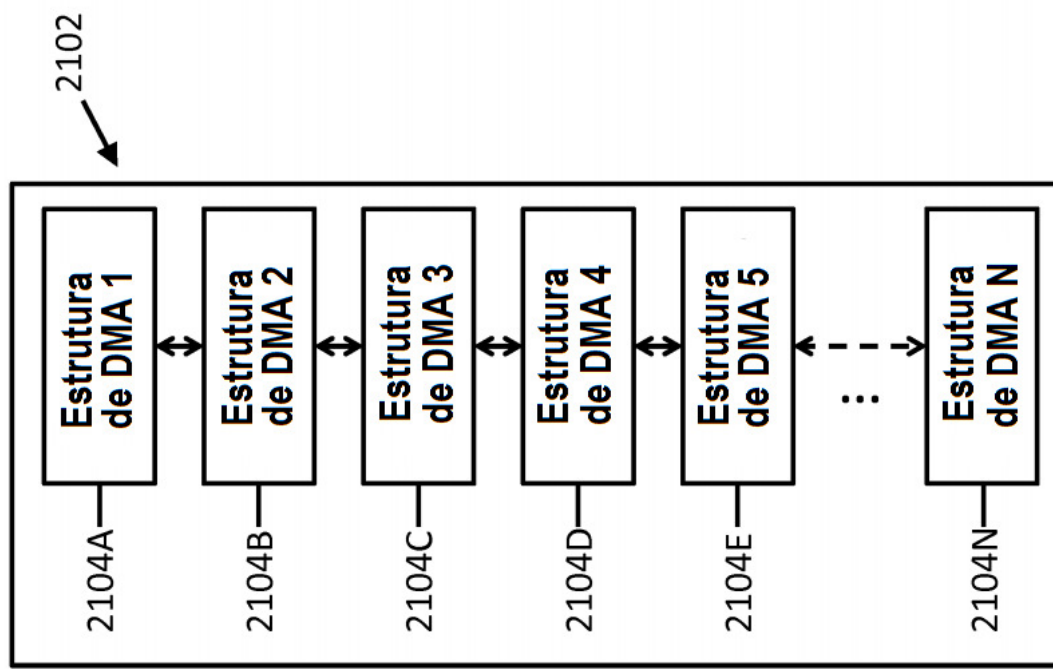
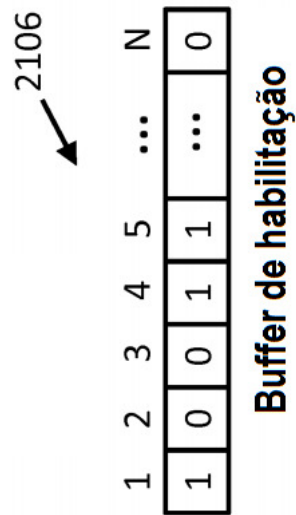


FIG. 21



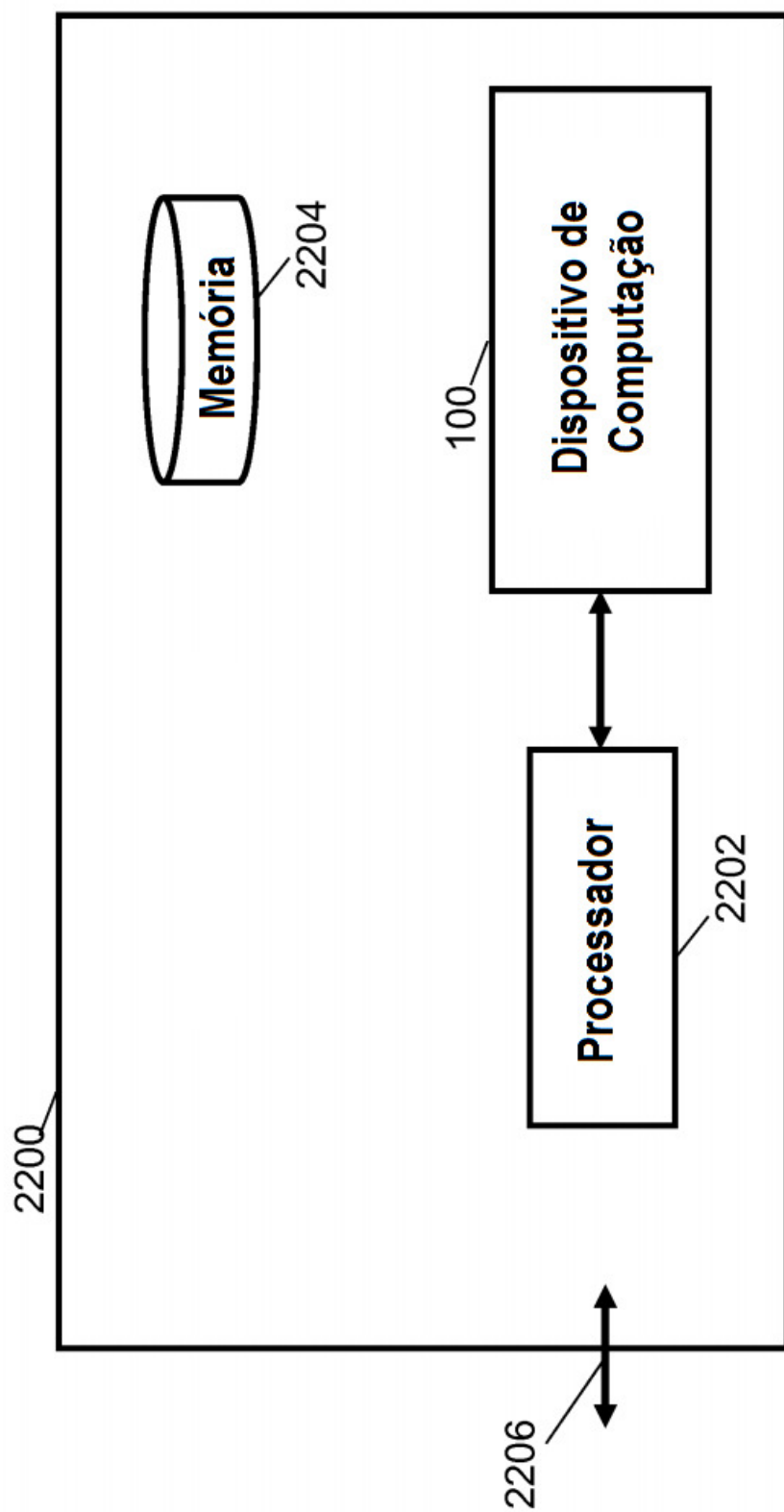


FIG. 22